64 PLUS 4

& AMIGA



GRUDZIEŃ 1991 ISSN 0867-3918 INDEKS 377112 CENA 8.000 zł

MIESIĘCZNIK UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE

Wesołych Świąt



OD REDAKCJI

Wadliwa dystrybucja „64 plus 4 & Amiga” przez przedsiębiorstwo RUCH jest przyczyną kłopotów, jakie mają czytelnicy chcący nabyć nasze pismo. Zdarza się, że otrzymujemy jako zwroty NIETKNIĘTE paczki zbiorcze. Tymczasem czytelnicy sygnalizują, że do wielu kiosków nie dociera ono wcale. Dlatego:

zapraszamy wszystkich chętnych do prowadzenia kolportażu „64 plus 4 & Amiga” (kluby, studia i sklepy komputerowe, księgarnie, osoby indywidualne itd.) do współpracy!

Oferujemy korzystne warunki!

Zainteresowanych prosimy o kontakt z działem dystrybucji pod adresem: Przedsiębiorstwo ABUK, 87-200 Wąbrzeźno, ul. 1 Maja 33.

NOWOŚĆ!

PUBLIC DOMAIN PACK dla COMMODORE 64 NA KASECIE!

Jeszcze w tym roku ukażą się cztery kasety!

Na naszych zestawach znajdują się programy, które do tej pory ukazały się na dyskach PDP, oraz wiele nowości!

Pojedyńcza kaseta kosztuje 30.000zł, wykupienie prenumeraty do końca roku (4 kasety) kosztuje tylko 110tys zł.

Wśród wszystkich, którzy zamówią komplet kaset (blankiet wpłaty zamieszczamy na str. 13) rozlosujemy 10 bezpłatnych prenumerat naszego pisma na rok 1992!

Na blankiecie prosimy dopisać PDP-taśma. Spis zawartości kaset wewnątrz numeru.

Przedsiębiorstwo ABUK S-ka z o.o. oferuje państwu **szybką i tanią obsługę reklamową.** Ogłoszenia drobne od osób indywidualnych (do 10 słów) przyjmujemy bezpłatnie. Większe - 1000 zł za słowo. Reklamy ramkowe (minimalny format - 20 cm^2): 1cm^2 ogłoszenia - **8000zł**, cała strona - **3,0 mln zł**; kolor - odpowiednio 100% drożej.

Ogłoszenia przyjmujemy za pośrednictwem poczty (nasz adres - patrz stopka redakcyjna). Treść ogłoszenia z określeniem formatu reklamy (ewentualnie zamówieniem koloru) prosimy nadsyłać listem poleconym wraz z odcinkiem wpłaty. Wpłat prosimy dokonywać za pomocą przekazu pieniężnego na konto Przedsiębiorstwa ABUK, Bank Polska Kasa Opieki SA Oddział w Bydgoszczy, konto nr : 5.09011-400522.7-136-11-111.0. Dołączenie do zamówienia odcinka wpłaty przyspieszy zamieszczenie reklamy .

64 PLUS 4

miesięcznik nr 12(14) grudzień 1991
cena 1 egz.: 8000 zł



Wydawca:
ABUK Spółka z o.o.
Redakcja nie ponosi odpowiedzialności za treść ogłoszeń.

Adres redakcji: Redakcja „64 plus 4” 85-166 Bydgoszcz 43 skrytka pocztowa 64
Redagują: Marcin Dudar, Sambor Kuźma, Paweł Sołtysiński, Waldemar Szczygieł (red. nacz.), Krzysztof Kobuz.
Okładka: Piotr Bartz.
Skład: ABUK
Druk: Z.P. POLRASTER 85-353 Bydgoszcz, ul. Orawska 19.

W numerze :



Z daleka i z bliska

* System 3 zapowiedział kilka nowych tytułów gier. Między nimi są Last Ninja 3 (gra dzięki której firma stała się znana) i Fuzball, platformowa zręcznościówka. „Pracujemy i inwestujemy w dobry rodzaj gier" mówi Adrian Cale, przedstawiciel firmy. Wszyscy wiedzą, że jesteśmy najlepsi na C-64, teraz naszym celem jest zajęcie pierwszej pozycji na Amidze. Rynek gier nasycił się, teraz gra musi nie tylko być dobra lecz musi się czymś wyróżniać spośród setek innych. Chcemy dać naszym klientom pewność, że wydając pieniądze na naszą grę będą zadowoleni. Teraz ludzie nie kupują już gier na podstawie impulsu - „O kupię sobie tę grę!". Coraz częściej jest to dobrze przemyślany krok. Chcemy mieć różnorodne i dobre jakościowo gry." Jak to dobrze, że my nie mamy takich problemów! I tak człowiek wszystko ma za darmo z pirackich kopii!

* Firma Roctec zapowiedziała najszczuplejszą stację dysków do Amigi. Mierząca 3/4 cala wysokości stacja nie tylko dobrze wygląda ale jest też solidnie zrobiona. Roctec już od wielu lat robi różne dodatki do Amigi lecz wcześniej sprzedawał je poprzez inne firmy. Teraz firma postanowiła robić to pod własną nazwą.

* Amiga 1000 powstaje z popiołów! Phoenix Microtechnologies z Australii wyprodukowała nową płytę dla Amigi 1000. Dzięki niej otrzymuje się: wewnętrzną stację dysków, gniazdo RS-232, gniazdo rozszerzenia i gniazdo video A2000, koprocesor matematyczny i 8Mb RAM'u. Przemiany dokonuje się bardzo łatwo wyjmując kilka kości ze starej płyty i wkładając je do nowej. Oczywiście należy zamienić również same płyty. Koszt: 450 funtów.

* Uwaga powstaje nowy rodzaj programów:ADDWARE. Nowość polega na tym, że za ukończenie gry będzie przyznana nagroda! Co więcej, celem Addware jest rozpowszechnianie programów więc kopiowanie jest ściśle wskazane! Przewidziane są również nagrody dla ludzi, którzy dostarczą grę temu pierwszemu, który ją ukończy.
Gdzie jest kruczek? Otóż nagrada przysługuje tylko i wyłącznie zarejestrowanemu posiadaczowi gry, co pociąga za sobą oczywiście pewne koszty. Pierwszą grą będzie Airmania. Jeśli chcecie uzyskać więcej danych to piszcie: Addware, PO Box 1991, Southampton, SO9 7HX.

* Było już wiele programów typu „napisz swoją własną przygodówkę". Firma Oxxi wypuściła jednak produkt, który bije na głowę poprzedników. „Aegis Visionary" pozwala na łączenie tekstu, grafiki i dźwięku w naprawdę dużych grach. Program obsługuje 64 tysięcy lokacji z 32 przedmiotami w każdej z nich; 64 tysięcy obiektów razem z postaciami, które nie są graczem lecz mogą ingerować w przebieg gry, 128 tysięcy zmiennych, 25 ekranów IFF, 25 dźwięków i 25 gadżetów na jednym ekranie w jednym czasie! Oczywiście program nie obrazi się jeśli podamy mu grafikę np. w HAM'ie! W zestawie jest również bardzo szybki kompiler i debbuger.

Uczymy się programować (cz.3)

Dzisiaj omówimy strings, czyli łańcuchy.

Granice pomiędzy poszczególnymi zakresami w pamięci użytkownika C-16 i Plus/4 są ruchome. Ich aktualne parzyste adresy znajdują się w kilku parach bajtów, tzw. wskaźnikach, na stronie zerowej księgi pamięci.

Na rys. 2 (patrz numer 10/91) wskaźnik ten jest zawsze podany obok odpowiedniej granicy.

Adres AD bajtu młodszego L i starszego H przedstawia się łącznie jako:

```
AD=L+256*H
```

lub rozdzielnie jako:

```
H=INT(AD/256),    L=AD-256*H.
```

Wszystkie zmienne proste A, A%, A$ i FNA zostają ułożone w tabeli zmiennych w grupy siedmiobajtowe, jedna za drugą, w kolejności występowania.

Komputer rozróżnia w bajtach nazwy czterech typów zmiennych, które mogą przyjmować wartości większe lub mniejsze niż 128. Większe bajty są drukowane na ekranie w inwersji. W przypadku zmiennych liczbowych można umieścić w grupie siedmiobajtowej nazwę i wartość, natomiast przy zmiennych łańcuchowych jest to niemożliwe, ponieważ powinny one obejmować 255 bajtów. W przypadku łańcuchów grupa siedmiobajtowa składa się z obu bajtów nazwy, trzech pozycji deskryptora łańcucha i dwóch zer wypełniających siedmiomiejscowe oznaczenie. Deskryptor łańcucha opisuje długość łańcucha i mówi gdzie zapamiętany jest jego tekst.

Pamięć łańcuchów rozpoczyna się od adresu 16374 (lub od 64768 w Plus/4) i wzrasta w kierunku zmiennych. W parze bajtów 53/54 znajduje się wskaźnik pomocniczy dla łańcuchów, który pokazuje zawsze koniec ostatniego łańcucha. Początek tych łańcuchów jest zawsze oznaczony przez wskaźnik 51/52.

Wskaźnik pomocniczy dla łańcuchów.

Aby prześledzić „obróbkę” łańcuchów należy całą pamięć zmiennych „ściągnąć” na ekran.

Osiąga się to rozkazem:

```
POKE46,12:POKE55,23:POKE56,13:CLR
```

Zmienne proste rozpoczynają się teraz od 256*12=3072, a więc w górnej linii ekranu.

Młodszy bajt, który znajduje się w linii 45, nie uległ zmianie. CLR ustala go rutynowo na 3. Koniec pamięci leży przy 23x256*13, a więc na 279 miejscu od początku ekranu. Jest to koniec siódmej linii. Od 8 linii mogą być wykonywane rozkazy.

Trzy dolne linie nie powinny być zapisane, ponieważ przesuwają pamięć zmiennych poza ekran.

Kończące CLR wzywa system pracy do umiejscowienia w nowych granicach.

Przełączmy się teraz w tryb małych liter przez równoczesne naciśnięcie SHIFT i Commodore key, skasujmy ekran przyciskiem Clear/Home i podajmy w linii 8 pierwszą zmienną:

```
ab$=„!”.
```

W górnym lewym rogu pojawia się grupa siedmiobajtowa, która rozpoczyna się bajtem nazwy AB.

Bajt 2, jak przy wszystkich zmiennych łańcuchowych przedstawiony jest w inwersji. Pozycja 6 i 7 jest wypełniona zerami, które w kodzie ekranu są reprezentowane przez klamry.

To, że rozkazy pisane małymi literami w naszej minipamięci pojawiają się jako duże, związane jest z przetworzeniem kodu ASCII w kod ekranu (zostało to wyjaśnione w pierwszej części).

Bajty od 3 do 5 tworzą trójmiejscowy deskryptor łańcuchów (czasami oznacza się tą nazwą całą grupę siedmiobajtową). Bajt 3 podaje długość łańcuchów; „a” oznacza „1”. Bajt 4 i 5 zawiera adres, pod którym w pamięci łańcuchów jest umieszczony tekst. W tym przypadku młodszy bajt t=20 i starszy m=13.

Jest to o trzy „numery” mniej niż granica pamięci, której młodszy bajt mamy zapisany w komórce 23. Tekst, wykrzyknik, musi więc rozpoczynać się trzy miejsca od końca 7 linii. Jest tam faktycznie i następują po nim oznaczenia e i 1.

Teraz zostanie podany nowy łańcuch, np.

```
cd$=„hans”.
```

Jak będzie wyglądać jego grupa siedmiobajtowa. Nazwa: C i w inwersji D; długość cztery, oznaczenie=d.

Starszy bajt adresu pozostaje m=13, a młodszy przesuwa się o sześć numerów w dół: n=14.

Tekst został wydrukowany z lewej strony obok pierwszego łańcucha. Pomiędzy tekstem i łańcuchem są dwie „1”.

Oba te oznaczenia, które w przypadku C-16, zostają „przyczepione” do każdego łańcucha, przedstawiają adres, który wskazuje na bajt długości przynależącego deskryptora łańcucha.

Trzeba to sprawdzić.

Chodzi tutaj, że tak powiem, o adres powrotny, który przyspiesza obróbkę łańcucha, a w szczególności tzw. „wywóz śmieci”. Co się właściwie dzieje, gdy zmienimy zmienną łańcuchową. Skasujemy teraz zmienną za pomocą CLR, a ekran przyciskiem Clear/Home.

Potem w linię 8 wpisuje się:

```
a$=„otto”
```

a następnie

```
a$=„max”
```

Do pamięci łańcuchów obok „Otto” zostaje wpisany nowy tekst „Max”.

W tabeli zmiennych nie pojawia się jednak żadna nowa grupa siedmiobajtowa, chociaż w starym deskryptorze łańcucha wpisano nowe wartości długości i adresu. Adres starego tekstu „Otto” jest zagubiony i chociaż Otto jeszcze jest w pamięci, to dla komputera jest już stracony. Co się dzieje z adresem powrotnym „zwłok łańcucha”? Został on przemieniony w „notatkę kasowania”, która składa się z informacji o długości (tutaj d=4) i oznaczeń szachownicy (kod ekranu 255). Rys. 4 pokazuje pamięć ze „śmietnikiem łańcuchów”. Dodatkowo zapisana jest pozycja wskaźnika.

Podczas przebiegu programu w pamięci gromadzi się mnóstwo zbytecznych łańcuchów, które zajmują cenne miejsce.

Nie ma problemu za śmieciami.

Komputer usuwa niepotrzebne łańcuchy w pierwszej kolejności, gdy brakuje pamięci, tzn. kiedy przez ciągłe pojawianie się nowych zmiennych, tabela zmiennych zderza się z pamięcią łańcuchów Teraz ta „kolekcja odpadków" wzywa „wywóz śmieci"- czyli procedurę systemową, która na podstawie deskryptora łańcucha sprawdza, które łańcuchy należy zachować. Potrzebne są przesuwane ku granicy pamięci, a „śmietnik" jest przepisywany.

Ta procedura sprawdzenia, przy której trzeba także zaktualizować wszystkie dane adresowe deskryptora, wymaga czasu. Wzrasta on w przypadku C-64 z kwadratem liczby ważnych łańcuchów i zabiera na 100 łańcuchów 80 sekund, za to dzięki adresom powrotnym w C-16 i Plus/4 trzeba już tylko na to 1,2 sekundy.

Ponadto czas obróbki wzrasta już tylko liniowo. Kiedy już także procedura śmietnikowa nie może zrobić wolnego miejsca w pamięci, komputer podaje sygnał błędu „out of memory".

Opróżnianie śmietnika nie powoduje tylko brak miejsca, ale to samo można zrobić rozkazem PRINT FRE(0).

Znowu czyści się pamięć i ekran przyciskiem CLR i CLR/Home i podaje następne zmienne:

a$=„pong":b$=„...."

Po naciśnięciu przycisku Return zmienia się 2 łańcuch:

b$=„ping".

Na kropeczkowym łańcuchu nie pokazuje się już teraz żaden deskryptor. Stał się on „odpadkowym łańcuchem". Jeśli poda się teraz PRINT FRE (0), to można obserwować jak nowy łańcuch przesuwa się na prawo. Ponadto do pamięci łańcuchów zostaje także wpisana liczba bajtowa z oznaczeniem kasowania. Jest to tzw. stała łańcuchowa, która nie jest przyporządkowana żadnej nazwie zmiennej. Inne stałe łańcuchowe tworzy się np. przez

PRINT„HAUS" lub IFA$=„MAUS"THEN...

Ich deskryptory nie zostają umieszczone w tabeli zmiennych, ale tylko przejściowo w stos dla łańcuchów czasowych, które znajdują się na stronie zerowej (adres od 25 do 33).

W dwóch przykładach zostanie wykazane, że przez zręczne programowanie, można zaoszczędzić i pamięć i czas. Dla wypełnienia wolnych miejsc w łańcuchach polecana jest często następująca pętla (tutaj dla lepszej czytelności jest wypełniona punktami).

a$=„x": do while len(a$):a$=a$+„.":loop

Niewiarygodne jaki śmietnik powoduje ta pętla przy wypełnianiu łańcuchów - ponad 200 zbytecznych bajtów. Teraz za pomocą PRINT FRE(0) przywołuje się „wywóz śmieci". Jedyny ważny łańcuch przesuwa się po tym do granicy pamięci, a wszystkie łańcuchy śmieci zostają zwolnione do przepisania.

Do identycznego wyniku dochodzi się stosując wskazówkę z maksymalnie 20 bajtami śmieci.

Wskazówka 13:
w celu wypełnienia łańcucha a$ na n pozycjach definiuje się „łańcuch pusty" sp$ z n miejscami, dodaje się go do a$ i używa funkcji LEFT$:
a$=left$(a$+sp$,n).

Im mniej zmiennych w pamięci, tym komputer może je szybciej odszukać.

Wskazówka 14:
powinno się wprowadzać jak najmniej nowych zmiennych.
Na zakończenie pętli z NEXT, może zostać użyta bez szkody dla następnej pętli, np. zmienna ruchoma.
We wskazówce 13 zmienna a$ została także użyta w nowym łańcuchu.

Teraz drugi przykład.

Co jest większe „Berlin",„ czy „Köln" ? Nawet jeśli Berlin ma więcej mieszkańców, jako łańcuch Köln jest większy.

Każda litera jest przedstawiana w pamięci przez jej wartość ASCII (liczba między 65 i 90) i dzięki tym liczbom komputer może ułożyć łańcuchy w porządku alfabetycznym. Bardzo znaną metodą porządkowania jest Bubble - Sort (listing 1). Polega ona na tym, że poczynając od A$(N), każdy człon A$(J) nieuporządkowanej listy jest porównywany z jego sąsiednim członem A$(J-1) (linia 170). Jeśli kolejność nie jest prawidłowa to oba człony zostają zamienione miejscami (linia 180), a w przeciwnym razie porównywane są następne elementy. Po pierwszym przejściu najmniejszy element wędruje na najniższą pozycję i nazywa się A$(1).

Z tego powodu konieczne jest N-1 porównań i prawie połowa tyle zamian. Szczególnie zamiany zabierają sporo czasu i pamięci, bowiem każdorazowo tworzą trzy zbyteczne bajty. W każdym następnym N-1 przejściu, jeden dalszy element trafia na właściwą pozycję. Dla uporządkowania 100 łańcuchów Bubble - Sort potrzebuje 108 sekund.

Godna polecenia jest inna metoda Shell - Sort (listing 2).

Potrzebuje ona N-1 przejść i maksymalnie N-1 porównań, ale każdy człon nie jest porównywany z sąsiednim lecz najmniejszym poprzednim, którego index jest przechowywany w zmiennej L. Po znalezieniu najmniejszego członu jego index jest zapamiętany w L (linia 170). Na koniec przebiegu jest zamieniany jednokrotnie najmniejszy element A$(L) i najniższy człon jeszcze nie uporządkowanej listy A$(J) (linia 190).

Shell - Sort porządkuje 100 łańcuchów w 50 sekund. Jest więc dwa razy szybsze od Bubble - Sort i tworzy znacznie mniej śmieci.

Ponieważ zapotrzebowanie czasu przy prostych procedurach porządkowania rośnie jak N do kwadratu, przy większej liczbie danych, potrzebe są szybsze procedury, które najczęściej pisane są w języku maszynowym.

Jeśli chcemy porządkować liczby za pomocą listing'u 1 lub 2, to koniecznie trzeba zamienić A$(J) i H$ na A(J) i H.

Można wprawdzie i bez tego wczytywać i porządkować liczby, ale wtedy ich kolejność się nie zgadza. Daje się to dowieść za pomocą operatorów porównań. Gdy twierdzenie jest prawdziwe komputer odpowiada - jeden, w przeciwnym razie - zero

np.

```
PRINT 3=2     odp.: 0, więc fałsz
PRINT 10>2    odp.: 1, prawda
```

ale

PRINT "10„>"2„ odp.: 0, fałsz.

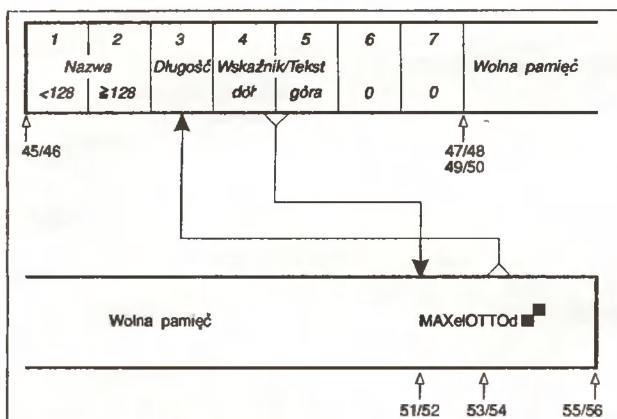
W przypadku łańcuchów, najpierw zostaną porównane dwa pierwsze oznaczenia, potem dwa następne i tak dalej.

```
100  REM BUBBLE - SORT
110  :
120  N=10:DIMA$(N)
130  FORI=1TON:READA$(I):NEXT
140  T=TI
150  FORI=2TON
160  FORJ=NTOISTEP-1
170  IFA$(J-1)A$(J)THEN190
180  H$=A$(J):A$(J)=A$(J-1):A$(J-1)=H$
190  NEXTJ,I
200  PRINT(TI-T)/60"S„
210  FORI=1TON:PRINTA$(I),:NEXT
220  :
230  DATA   ULM,BERLIN,MAINZ,KOELN,ESSEN
240  DATA   KIEL,TRIER,KASSEL,FULDA,BONN
```

Listing 1: procedura Bubble - Sort.

```
100  REM SHELL - SORT
110  :
120  N=10:DIMA$(N)
130  FORI=1TON:READA$(I):NEXT
140  T=TI
150  FORI=1TON-1:L=I
160  FORJ=1TON
170  IFA$(J)A$(L)THENL=J
180  NEXTJ
190  H$=A$(L):A$(L)=(I):A$(I)=H$
200  NEXTI
210  PRINT(TI-T)/60"S„
220  FORI=1TON:PRINTA$(I),:NEXT
230  :
240  DATA   ULM,BERLIN,MAINZ,KOELN,ESSEN
250  DATA   KIEL,TRIER,KASSEL,FULDA,BONN
```

Listing 2: alternatywna procedura porządkowania.



Rys. 4 Tak wygląda pamięć z „śmietnikiem łańcuchów"

Opracowano na podstawie RUN 4/87.

JAK ZROBIĆ DEMO (9).

W dzisiejszym odcinku zajmiemy się czymś nowym. Jak powszechnie wiadomo, w Commodore 64 w przypadku robienia „płynącyh napisów" zadanie to ułatwia specjalizowany układ VIC, który przejmuje na siebie część problemów związanych ze scrollingiem. Autorzy podobnych procedur na inne komputery (np. na Spectrum, Amstrad) nie mają takich możliwości i pozostaje im tylko „mielenie pamięci" przy użyciu procesora. Chodzi tu między innymi o to, że wrażenie płynności uzyskuje się poprzez odpowiednie rozkazy procesora, które przesuwają zawartość poszczególnych bajtów o jeden bit w wybraną stronę (w lewo lub w prawo), co odpowiada mnożeniu lub dzieleniu wartości danego bajtu przez 2. Nie mamy więc tu do czynienia z kodami znaków, które potem VIC wyświetla na ekranie lecz z obrazem graficznym, składającym się z zapisanych tam definicji graficznych poszczególnyh liter. Z taką sytuacją spotykamy się na Commodore 64 w przypadku pisania procedury płynnego scroll'a utworzonego przez graficzne przesuwanie liter w ustawionych obok siebie sprite'ach.

Nasza procedura będzie to ralizować wraz z efektownym cieniem (tzn. będą to dwa scroll'e, z których jeden będzie kopią drugiego z tym, że trochę przesuniętą o pewien wektor tak, aby stworzyć wrażenie cienia). Przedstawiony listing należy wpisać używając dowolnego assemblera, a najlepiej TurboAssemblera.

Jednocześnie chciałbym z przykrością poinformować, że jest to już ostatni odcinek działu „Jak zrobić własne demo". Mam jednak nadzieję, że przedstawione tutaj pomysły zachęcą Was do własnych eksperymentów.

```
        *= $1000 ;ADRES STARTU

SHAD1   = $05     ;PARAM. X PRZESUNIECIA CIENIA
SHAD2   = $04     ;PARAM. Y PRZESUNIECIA CIENIA

        SEI
        LDA #$7F
        STA $DCOD
        LDX #$00
        STX $DC0E
        INX
        STX $D01A
        DEX
        ORA #$80
        STA $D015          ;WLACZENIE WSZYSTKICH
                           ;SPRITES
        STX $D01C          ;SPRITES-MULTICOLOR OFF
        STA $D01D          ;EXPAND X ON
        STX $D017          ;EXPAND Y OFF
        STX $D01B          ;PRIORYTET TLA - OFF
        TXA
LP1     STA $0F00,X        ;CZYSZCZENIE PAMIECI
                           ;SPRITES
        INX
        BNE LP1
        LDY #$0F00/64      ;USTAWIENIE ADRESOW
                           ;DANYCH DLA SPRITES
        STY $07F8
        STY $07FC
        INY
        STY $07F9
        STY $07FD
        INY
        STY $07FA
        STY $07FE
        INY
        STY $07FB
        STY $07FF
        TXA
        LDX #$C0
LP0     STA $03C0,X        ;CZYSZCZENIE PAMIECI
                           ;KOLEJNEGO ZNAKU
        INX
        BNE LP0
        STX $D010          ;USTAWIANIE POLOZENIA
                           ;SPRITES
        LDA #$58
        STA $02            ;KOMORKA POMOCNICZA
LP2     LDA $02
        STA $D000,X
        CLC
        ADC #$30
        STA $02
        CLC
        ADC #SHAD1
        STA $D008,X
        LDA #$D8
        STA $D001,X
        CLC
        ADC #SHAD2
        STA $D009,X
        INX
        INX
        CPX #$08
        BNE LP2
        JSR INIT           ;USTALANIE PARAMETROW
                           ;TEKSTU
        LDA #<IRQ
        LDX #>IRQ
        LDA #$1B
        STA $D011          ;USTALANIE NUMERU LINII
                           ;DLA PRZERWANIA
        LDA ##$F8
        STA $D012
        CLI
        RTS                ;POWROT DO BASIC

IRQ     JSR SCROLL         ;WYWOLANIE SCROLL'A
                           ;W PRZERWANIU
        INC $D019          ;POTWIERDZENIE
                           ;NASTEPNEGO PRZERWANIA
        JMP $EA31          ;SKOK DO ROM

COUNTER .BYTE 0            ;KOMORKA POMOCNICZA

INIT    LDA #$01
        STA COUNTER
TXSET   LDA #<TXT          ;USTALANIE ADRESU
                           ;TEKSTU
        STA $20
        LDA #>TXT
        STA $21
        RTS

SCROLL  JSR LROL           ;WYWOLANIE PROCEDURY
                           ;PRZEWIJANIA
        BEQ SCR1           ;TAK - NASTEPNA LITERA
        RTS
SCR1    LDY #$08
        STY COUNTER        ;USTALENIE LICZNIKA
        TAY
```

```
SCR2    LDA  ($20),Y      ;POBIERANIE  KOLEJNEGO
                          ;ZNAKU
        AND  #$3F         ;Z KODU ASCII DO
                          ;KODOW  EKRANOWYCH
        BNE  SCR3
        JSR  TXSET        ;KONIEC  TEKSTU  -
                          ;USTAW  PONOWNIE
        BNE  SCR2         ;POBIERZ  PIERWSZY  ZNAK
SCR3    STY  $03          ;OBLICZANIE  ADRESU
                          ;DANYCH  DLA  ZNAKU
        ASL  A
        ROL  $03
        ASL  A
        ROL  $03
        ASL  A
        ROL  $03
        STA  $02
        LDA  $03
        CLC
        ADC  #$D0
        STA  $03
        LDA  #$33
        STA  $01          ;PRZELACZENIE  ROM  NA
                          ;GENERATOR  ZNAKOW
        LDX  #$00
SCR4    LDA  ($02),Y      ;PRZEPISYWANIE  DANYCH
                          ;DO  POLA  NASTEPNEGO
        STA  $03C0,X      ;ZNAKU
        INX
        INX
        INX
        INX
        INX
        INX
        INY
        CPY  #$08
        BNE  SCR4
        INC  $20          ;PRZYGOTUJ  POBRANIE
                          ;NASTEPNEGO  ZNAKU
        BNE  SCR5
        INC  $21
SCR5    LDA  #$37
        STA  $01          ;STANDARDOWY  ROM
        RTS

LROL    LDX  #$00         ;PROCEDURA
                          ;PRZESUWANIA  BITOW
        ASL  $03C0,X      ;KOPIOWANIE Z POLA
                          ;NASTEPNEGO  ZNAKU
        ROL  $0FC2,X
        ROL  $0FC1,X
        ROL  $0FC0,X
        ROL  $0F82,X
        ROL  $0F81,X
        ROL  $0F80,X
        ROL  $0F42,X
        ROL  $0F41,X
        ROL  $0F40,X
        ROL  $0F02,X
        ROL  $0F01,X
        ROL  $0F00,X
        TXA
        CLC
        ADC  #$06
        TAX
        CPX  #$30
        BNE  LROL+2
        DEC  COUNTER
        RTS

TXT     .TEXT „COMMODORE 64  SPRITE-SCROLLER
                          WITH A SHADOW!!"
        .BYTE 0           ;KONIEC  TEKSTU
```

Paweł Sołtysiński

Turbo Assembler
Help

Na czerwcowym dysku Public Domain był rozprowadzany bardzo popularny na Zachodzie Turbo Assembler v.5, który przez swoją wygodę i wszechstronność w pełni sobie na ową popularność zasługuje. Tak się jednak złożyło, że dotarł on do Polski bez jakichkolwiek instrukcji. Ja też miałem na początku sporo problemów z jego obsługą i trwało to aż do dnia, w którym wziąłem go „na warsztat" i przy pomocy monitora prześledziłem jego procedury.

Na tej podstawie mogę teraz stwierdzić, że klawisz „strzałka w lewo" jest początkiem wydawania komend, które wybiera się następującymi po nim znakami:

— A - pisanie znakami sterującymi; wyjście: „strzałka w lewo"
— B - rozkazy dla zdefiniowanych bloków tekstu:
— write - zapis na dysk lub wydruk po podaniu znaku „?" zamiast nazwy
— C - „zimny start"
— D - rozkazy dla stacji dysków
— E - ładowanie zbioru typu SEQ w miejsce wskazane kursorem
— F - find (szukaj); „strzałka w lewo" + H - następne wyrażenie
— Shift+F - memory fill
— G - idź do zdefiniowanego wcześniej położenia
— K - redefinicja klawiszy funkcyjnych
— Shift+K - key-click on/off
— L - ładowanie tekstu programu w postaci upakowanej
— Shift+L - ładowanie danych z dysku pod wskazany adres
— M - markowanie położeń 0-9, s, e
— N - idź do wskazanej linii
— O - zmiana barw obrazu
— I - czyszczenie pamięci
— R - replace; „strzałka w lewo"+T - zamieniaj
„strzałka w lewo" +H - omiń
„strzałka w lewo" +Y - zamień wszystkie wyrazy
— S - nagraj tekst w postaci upakowanej
— U - listuje wszystkie etykiety; na ekran: „*"
na drukarkę: „?"
na dysk: podaj nazwę
— Y - testuj program
— @ - diskstatus
— * - disk directory
— space - kasowanie linii
— delete - likwidowanie linii
— insert - tworzenie nowej linii
— 1 - wyjście do BASIC
— 2 - tworzy linię ze znaków „minus"
— 3 - assemblacja do pamięci
— 4 - assembler listing; na ekran: „*"
na drukarkę: „?"
na dysk: podaj nazwę
— 5 - assemblacja na dysk
— 6 - interpretowanie danych z pamięci jako tekst prog ramu
— 7 - ustalanie tabulatorów
— cursor-up - przewijanie po 200 linii w górę
— cursor-down - przewijanie po 200 linii w dół
— : - lista wszystkich zamarkowanych linii
— ; - kasowanie zamarkowań
— + - dodawanie liczb dwubajtowych
— - - odejmowanie liczb dwubajtowych
— „strzałka w górę" - zapamiętanie linii do bufora
— znak funta - odtworzenie zapamiętanej w buforze linii
— return - ustawia kursor w pozycji 0 od lewej strony

Mam nadzieję, że to zestawienie się przyda...

Paweł Sołtysiński

CZY 128D =128D?

Powszechnie wiadomo, że 128D w obudowie blaszanej różni się od pozostałych 128D. Na czym polegają te różnice?

Obok zmian hardware'owych (nowa płyta główna, 64Kb VDC-RAM, nowy chip sterownika stacji i blaszana obudowa) dokonano także modyfikacji systemu operacyjnego.

Zmiany te miały na celu zmniejszenie kosztów produkcji (np. montaż sterownika stacji metodą powierzchniową) oraz usunięcie niektórych błędów systemowych. Nie wszystkie wady udało się usunąć (np. brak przystosowania do rozszerzenia pamięci VDC).

Po włączeniu komputera z 64Kb dostępne są tylko 32Kb pamięci VDC. Dopiero przełączając 4 bit w 28 rejestrze VDC można uzyskać dostęp do pełnych 64Kb.

Zupełnie odmiennie umieszczono system operacyjny w pamięciach ROM. W „starym" 128D w układzie U32 mieści się system operacyjny C-64, w U33-górna, a w U34 dolna część C-128 Basic ROM.

U35 zawiera system operacyjny C-128 i CP/M LOADER.

W wersji „blaszanej" U32 zawiera system operacyjny trybu C-64, C-128 oraz CP/M LOADER, a U34 - BASIC C-128.

W wersji tej użyto także nowych układów pamięci RAM (64Kx4) - redukując w ten sposób ilość kości z 16 sztuk do czterech. Nowy jest też układ procesora obrazu (nie jest kompatybilny ze swoim poprzednikiem).

Adres	C128	C128 DB
41F0	35	36
4D70/71/72	8E 0C 12	20 D1 7E
5C87/88	8F 5C	8E 7E
5ED7/D8	C8 D0	E6 61
5ED9?DA	F8 E6	D0 02
5EDB/DC	62 E6	E6 62
5EDD/DE/DF	27 AD 11	20 94 7E
5EE0/E1	12 C5	90 F0
5EE2/E3/E4	27 B0 ED	EA EA EA
5EF8/F9/FA	4C 37 4D	60 00 00
5F02	0C	2E
66A5/A6	20 32	B0 11
66A7/A8/A9	9E 20 86	AD 54 11
66AA/AB/AC	03 D0 03	8D 56 11
66AD/AE/AF	4C FB 9B	AD 55 11
66B0/B1	20 86	24 D8
66B2/B3	03 C9	10 04
66B4/B5/B6	2C F0 05	0E 56 11
66B7	C9	2A
66B8/B9/BA	A4 F0 01	8D 57 11
66BB/BC	60 48	A2 23
66BD/BE/BF	20 80 03	20 4A 9D
66C0/C1/C2	A2 04 20	EA EA EA
66C3/C4	70 9E	EA EA
6839/3A	8D 92	82 7E
6845/46	0C C0	88 7E
6860/61	0C C0	88 7E
6F45/46/47	8D 2D 12	20 A6 7E
6FF9/FA/FB	2F B2 83	0F 0C 45
6FFC/FD/FE	99 FC B1	Bf 7D 83
6FFF/00/01	BD 25 EF	D6 79 73
7002/03/04	20 BE D1	7C 97 43
7005/06/07	4C 50 55	43 47 4B
7008/09/0A	5A 5F 65	4F 54 59
700B/0C/0D	6B 72 78	5E 64 6A
700E/0F/10	80 87 8F	70 77 7E
7A31/32	B7 FF	51 92
7E82/83/84	FF FF FF	20 45 A8
7E85/86/87	FF FF FF	4C F0 FF
7E88/89/8A	FF FF FF	20 45 A8
7E8B/8C/8D	FF FF FF	4C 0C C0
7E8E/8F/90	FF FF FF	20 8F 5C
7E91/92/93	FF FF FF	4C 86 03
7E94/95	FF FF	E6 26
7E96/97	FF FF	D0 02
7E98/99	FF FF	E6 27
7E9A	FF	38
7E9B/9C	FF FF	A5 26
7E9D/9E/9F	FF FF FF	ED 10 12
7EA0/A1	FF FF	A5 27
7EA2/A3/44	FF FF FF	ED 11 12
7EA5	FF	60
7EA6/A7/A8	FF FF FF	2C 03 0A
7EA9/AA	FF FF	30 04
7EAB/AC/AD	FF FF FF	BD 05 70
7EAE	FF	60
7EAF/B0/B1	FF FF FF	BD B9 7E
7EB2/B3/B4	FF FF FF	8D 2D 12
7EB5/B6/B7	FF FF FF	BD C5 7E
7EB8	FF	60
7EB9/BA/BB	FF FF FF	9C BF 22
7EBC/BD/BE	FF FF FF	C8 B4 EB
7EBF/C0/C1	FF FF FF	71 4C 7F
7EC2/C3/C4	FF FF FF	12 98 68
7EC5/C6/C6	FF FF FF	45 49 4E
7EC8/C9/CA	FF FF FF	52 57 5C
7ECB/CC/CD	FF FF FF	62 68 6E
7ECE/CF/D0	FF FF FF	75 7C 83
7ED1/D2/D3	FF FF FF	8E 0C 12
7ED4/D5	FF FF	A2 1B
7ED6/D7	FF FF FF	86 18
7ED8	FF	60
7FC0/C1/C2	FF FF FF	28 43 29
7FC3/C4/C5	FF FF FF	31 39 38
7FC6/C7/C8	FF FF FF	36 20 43
7FC9/CA/CB	FF FF FF	4F 4D 4D
7FCC/CD/CE	FF FF FF	4F 44 4F
7FCF/D0/D1	FF FF FF	52 45 20
7FD2/D3/D4	FF FF FF	45 4C 45
7FD5/D6/D7	FF FF FF	43 54 52
7FD8/D9/DA	FF FF FF	4F 4E 49
7FDB/DC/DD	FF FF FF	43 53 2C
7FDE/DF/E0	FF FF FF	20 4C 54
7FE1/E2/E3	FF FF FF	44 2E 20
7FE4/E5/E6	FF FF FF	41 4C 4C
7FE7/E8/E9	FF FF FF	20 52 49
7FEA/EB/EC	FF FF FF	47 48 54
7FED/EE/EF	FF FF FF	53 20 52
7FF0/F1/F2	FF FF FF	45 53 45
7FF3/F4/F5	FF FF FF	52 56 45
7FF6/F7	FF FF	44 2E
7FFC/FD/FE	FF FF FF	EF 8D 01
7FFF	FF	61
8323	10	08
839C	10	08
8A3C	55	5B
8A4B	55	5B

8A57	38	EA
98BB/BC/BD	38 6E 25	20 6E AA
98BE	01	EA
9F18/19	12 88	C0 6D
A153	38	EA
A155/56	75 92	B2 A3
AA6E/6F/70	FF FF FF	20 D3 42
AA71/72	FF FF	C9 2C
AA73/74	FF FF	D0 07
AA75/76/77	FF FF FF	AD 33 01
AA78/79/7A	FF FF FF	20 EB 98
AA7B	FF	C8
AA7C	FF	38
AA7D/7E/7F	FF FF FF	6E 25 01
AA80	FF	60
BFC0/C1/C2	FF FF FF	28 43 29
BFC3/C4/C5	FF FF FF	31 39 38
BFC6/C7/C8	FF FF FF	36 20 43
BFC9/CA/CB	FF FF FF	4F 4D 4D
BFCC/CD/CE	FF FF FF	4F 44 4F
BFCF/D0/D1	FF FF FF	52 45 20
BFD2/D3/D4	FF FF FF	45 4C 45
BFD5/D6/D7	FF FF FF	43 54 52
BFD8/D9/DA	FF FF FF	4F 4E 49
BFDB/DC/DD	FF FF FF	43 53 2C
BFDE/DF/E0	FF FF FF	20 4C 54
BFE1/E2/E3	FF FF FF	44 2E 20
BFE4/E5/E6	FF FF FF	41 4C 4C
BFE7/E8/E9	FF FF FF	20 52 49
BFEA/EB/EC	FF FF FF	47 48 54
BFED/EE/EF	FF FF FF	53 20 52
BFF0/F1/F2	FF FF FF	45 53 45
BFF3/F4/F5	FF FF FF	52 56 45
BFF6/F7	FF FF	44 2E
BFFC/FD/FE	FF FF FF	C8 CD 01
BFFF	FF	C5
C106/07/08	A2 0B BD	20 81 FE
C109/0A	6F C0	A9 00
C10B/0C/0D	8D 34 03	8D 3F 03
C10E/0F/10	CA 10 F7	20 87 FC
C6CA/CB/CC	BD 00 10	4C 00 CF
CF00/01	FF FF	A5 D1
CF02/03	FF FF	D0 12
CF04/05/06	FF FF FF	BD 00 10
CF07/08	FF FF	85 D1
CF09/0A	FF FF	A9 00
CF0B	FF	18
CF0C	FF	CA
CF0D/0E	FF FF	30 05
CF0F/10/11	FF FF FF	7D 00 10
CF12/13	FF FF	90 F8
CF14/15	FF FF	85 D2
CF16/17/18	FF FF FF	4C C4 C6
CF60/81/82	FF FF FF	AD 06 D5
CF83	FF	48
CF84/85/86	FF FF FF	5D F0 F7
CF87/88	FF FF	29 3F
CF89/8A/89	FF FF FF	5D F0 F7
CF8C/8D/8E	FF FF FF	8D 06 D5
CF8F/90/91	FF FF FF	BD F0 F7
CF92	FF	AA
CF93	FF	08
CF94	FF	78
CF95/96/97	FF FF FF	20 F0 03
CF98	FF	28
CF99	FF	68
CF9A/9B/9C	FF FF FF	8D 06 D5
CF9D	FF	60
CFC0/C1/C2	FF FF FF	28 43 29
CFC3/C4/C5	FF FF FF	31 39 38
CFC6/C7/C8	FF FF FF	36 20 43
CFC9/CA/CB	FF FF FF	4F 4D 4D

CFCC/CD/CE	FF FF FF	4F 44 4F
CFCF/D0/D1	FF FF FF	52 45 20
CFD2/D3/D4	FF FF FF	45 4C 45
CFD%/D6/D7	FF FF FF	43 54 52
CFD8/D9/DA	FF FF FF	4F 4E 49
CFDB/DC/DD	FF FF FF	43 53 2C
CFDE/DF/E0	FF FF FF	20 4C 54
CFE1/E2/E3	FF FF FF	44 2E 20
CFE4/E5/E6	FF FF FF	41 4C 4C
CFE7/E8/E9	FF FF FF	20 52 49
CFEA/EB/EC	FF FF FF	47 48 54
CFED/EE/EF	FF FF FF	53 20 52
CFF0/F1/F2	FF FF FF	45 53 45
CFF3/F4/F5	FF FF FF	52 56 45
CFF6/F7	FF FF	44 2E
CFF9/FA/FB	FF FF FF	00 F3 91
CFFC/FD/FE	FF FF FF	38 EA 01
CFFF	FF	EA
E159/5A/5B	8D 37 0A	20 55 FC
E192/93	DC E1	67 FC
E337	27	26
EF19/1A/1B	85 D6 A5	20 40 FC
EF1C/1D	E7 85	A5 E7
EF1E/1E	EA 4C	85 EA
EF20/21	09 C0	B0 F0
F09D/9E	0A B0	30 04
F09F/A0/A1	03 20 55	20 55 E7
FC22	51	D1
FC40/41	FF FF	05 D6
FC42/43	FF FF	85 D6
FC44/45	FF FF	A5 EB
FC46/47/48	FF FF FF	8D 30 0A
FC49		60
FC4A/4B	FF FF	A5 AE
FC4C/4D	FF FF	A4 AF
FC4E/4F	FF FF	85 AC
FC50/51	FF FF	84 AD
FC52/53/54	FF FF FF	4C 33 F5
FC55/56/57	FF FF FF	8D 37 0A
FC58/59/5A	FF FF FF	8D 11 0A
FC5B/5C/5D	FF FF FF	8D 12 0A
FC5E/5F/60	FF FF FF	8D 13 0A
FC61/62	FF FF	A2 08
FC63/64/65	FF FF FF	8E 10 0A
FC66	FF	60
FC67/68	FF FF	A0 00
FC69/6A	FF FF	A9 7F
FC6B/6C/6D	FF FF FF	8C 00 D6
FC6E/6F/70	FF FF FF	8D 01 D6
FC71/72/73	FF FF FF	4C DC E1
FE81/82	FF FF	A5 01
FE83/84	FF FF	29 40
FE85/86	FF FF	49 40
FE87	FF	4A
FE88	FF	4A
FE89/8A	FF FF	85 D3
FE8B	FF	60
FF51/52	A5 F7	60 CF
FF80	D0	01

Opracowano na podstawie 64'er nr 7/90.

Od czego zacząć?

Zacząłem otrzymywać wiele listów od osób, które zainteresowały się programowaniem swojego Commodore 64. Co bardziej ambitni planują tworzenie własnych programów demonstracyjnych, zapytują w listach o konkretne rozwiązania, przysyłają próbki swoich prac. Trudno jest nam na wszystkie te pytania odpowiedzieć.

Jest jednak takie jedno, dość często się powtarzające pytanie: jakie programy użytkowe należy odnaleźć i dołączyć do swoich zbiorów by uzyskać pewien sprawny pakiet narzędzi na każdą okazję. Ponieważ pytanie to kierują głównie nowo powstające grupy, wobec tego omówię oprogramowanie dla każdego z członków takiej samowystarczalnej grupy:

- Programista (czyli Koder): potrzebuje narzędzi do pisania innych programów, a więc assemblerów. z czego najlepszy będzie TurboAssembler v.5 lub MagicAssembler. Przydało by się także parę dobrych monitorów pamięci, które rozpoznają zastrzeżone rozkazy procesora i tu polecam Starmon'a i MagicMonitor. Z dodatkowych pomocy: posiadanie dobrej mapy pamięci i lepszego cardridge'a (w zasadzie w grę wchodzi tylko Action Replay). Wszystkie Black Box'y i inne im podobne odpadają. Pozostała sprawa kompresorów - należy szukać CruelCruncher'a, PowerPacker'a, FastCruel'a, CSLinker'a, Charblaster'a, Zipper'a - werjsę 5-tą oraz KompressMaster'a.
- Muzyk (o ile takiego posiadamy) powinien być zainteresowany edytorami muzycznymi. Wystarczyć tu powinien Voicetracker, Romuzak, Graffiti Music Maker, czy Music Assembler. Mogą być także różne inne, w każdym razie jedna wskazówka — gdy ktoś na giełdzie będzie Wam wciskał jakiś edytor muzyczny z klasycznym zapisem dźwięków na pięciolinii, to dajcie sobie z tym spokój — taki program nie nadaje się do niczego oprócz, być może, zastosowań edukacyjnych.
- Grafik (o ile takiego posiadamy) powinien być zaopatrzony w kilka programów graficznych, w tym dowolny do definiowania sprite'ów, do edycji kształtów liter w kilku rozmiarach (najlepszy jest FontEditor węgierskiej grupy FACES), programy do grafik w trybie multicolor, a więc bezapelacyjnie najlepszy AmicaPaint autorstwa Olivera Stiller'a, no i może jeszcze Advanced Art Studio. Przydałoby się namówić grupowego kodere na zrobienie programu do konwersji obrazków z jednego edytora na drugi. Ważne jest także posiadanie dobrego edytora do edycji grafiki zrobionej na zestawie znaków, z możliwością przeliczania obrazów graficznych np. z AdvArtStudio w generator znaków. Wymogi te spełnia MultiDrawer autorstwa Silver Dream'a, lub z trochę gorszym skutkiem, LogoPainter v.3+mojego autorstwa.
- Swapper (czyli grupowy „wymieniacz" programów) powinien posiadać dobre programy kopiujące, z czym nie ma chyba większego problemu: najlepszy jest zestaw FastHack'em 4.0, zawierający nibbler'y, backup'y i programy do kopiowania pojedynczych zbiorów. Kopiującym większe ilości dysków polecałbym zamontowania sobie tzw. łączki równoległej, łączącej port w stacji dysków z user portem w naszym Commodore. Przy użyciu odpowiednich programów (np. 35-tracks copier by Fall Guys), które czas kopiowania dyskietki skracają do około 15 sekund. Przydała by się także mała baza danych pod nazwą „co i do kogo wysłałem, a od kogo oczekuję na przesyłkę" - rzecz po jakimś czasie prawie nieodzowna.

Paweł Sołtysiński

WSZYSTKICH ZAINTERESOWANYCH NABYCIEM ZALEGŁYCH NUMERÓW

„64 plus 4 & AMIGA"

INFORMUJEMY, ŻE POSIADAMY JESZCZE OGRANICZONĄ ILOŚĆ NUMERÓW

OD LISTOPADA 1990R. DO WRZEŚNIA 1991R.

ZAMÓWIENIA PROSIMY KIEROWAĆ NA ADRES :
Przedsiębiorstwo ABUK sp. z o.o.,
87-200 Wąbrzeźno,
ul. 1 Maja 33.

(Pod tym adresem mieści się dział kolportażu - tam też prosimy przesyłać wszelką korespondencję dotycząca kolportażu czasopisma, dyskietek, taśm itd. Adres redakcji się nie zmienił - patrz stopka.)

ZAMÓWIONE NUMERY PRZEŚLEMY ZA ZALICZENIEM POCZTOWYM.

W związku z pojawiającymi się kłopotami w dystrybucji oferowanych przez nas dyskietek i taśm (wynikającymi z nieczytelnego bądź niekompletnego wypełnienia blankietów wpłat) przedstawiamy obok specjalny druk. Blankiet ten może służyć jako zamówienie i dowód wpłaty dla wszystkich oferowanych przez nas usług: sprzdaż dyskietek i taśm PDP, Voicetracker'a, zamówienie ogłoszeń itd.

REDAKCJA

UWAGA!

Informujemy, że od 1 stycznia 1992r cena 1 egzemplarza „64 plus 4" w prenumeracie wynosi 10.000zł.

Przypominamy, że prenumeratę można zawrzeć na okres nie krótszy niż trzy miesiące DO KOŃCA ROKU KALENDARZOWEGO, tzn do końca 1992r..
Na **CZYTELNIE** wypełnionych blankietach wpłat prosimy dopisać **„PRENUMERATA" oraz okres jej trwania.**
Wpłaty prosimy przesyłać wykorzystując zamieszczony obok blankiet.
Dla wszystkich, którzy zdecydują się wykupić prenumeratę roczną - specjalna okazja - patrz 2 str.

Odcinek dla wpłacającego
Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)
na rachunek:
**Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-136-11-111.0.**
Opłata
zł..........

Odcinek dla Banku
Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)
na rachunek:
**Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-136-11-111.0.**
Opłata
zł..........

Odcinek dla posiadacza rachunku
Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)
na rachunek:
**Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-136-11-111.0.**
Opłata
zł..........

Odcinek dla Poczty
Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)
na rachunek:
**Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-136-11-111.0.**
Opłata
zł..........

TREŚĆ ZAMÓWIENIA: ....

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA: ....

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA: ....

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA: ....

W związku z znacznym wzrostem opłat za przesyłki pocztowe, mając na uwadze dobro naszych czytelników sugerujemy zamawianie numerów zaległych naszego czasopisma na nieco innych zasadach.

Koszty przesyłki za tzw. zaliczeniu pocztowym przedstawia tabela (rubryki 2, 3, i 4). Wynika z niej, że koszty przesłania dwóch pierwszych numerów „64 plus 4" są większe niż ich wartość! Zamawiając numery wartości 6.000 zł zmuszeni jesteście dopłacić poczcie jeszcze 8.000 zł!

Zupełnie inaczej przedstawia się sytuacja w tabelach 5, 6 i 7. Wpłata kwoty z rubryki nr 7 na nasze konto (wraz z czytelną adnotacją, których numerów dotyczy, umieszczoną na wszystkich odcinkach blankietu) powoduje, że otrzymujecie przesyłkę bez kosztów pobrania!

Proponujemy abyście zaległe numery naszego pisma zamawiali w sposób następujący: wykorzystując tabelę - rubryki 1, 2, 5 i 7 - wyliczyli kwotę wpłaty, a następnie przesłali ją na nasze konto. Po otrzymaniu wpłaty natychmiast realizujemy przesyłkę!

Uwaga: dla dokonania wpłaty prosimy wykorzystać blankiet zamieszczony na tej stronie. Wszystkie dane prosimy pisać czytelnie i - zgodnie z rubrykami na blankiecie - **dziękujemy!**

Numery	Za pobraniem			Wpłata na konto		
	Wart. pisma	Koszt wysyłki	Razem	Wart. pisma	Porto	Razem
1	2	3	4	5	6	7
XI,XII	6.000,-	8.000,-	14.000,-	6.000,-	1.000,-	7.000,-
XI,XII,I	11.000,-	8.500,-	19.500,-	11.000, 1.500,-	12.500,-	
XI,XII,I,II	16.000,-	8.500,-	24.500,-	16.000,-	1.500,-	17.500,-
XI,XII,I,II,III	21.000,-	8.500,-	29.500,-	21.000,-	1.500,-	22.500,-
XI,XII,I-IV	26.000,-	10.000,-	36.000,-	26.000,-	2.000,-	28.000,-
XI,XII,I-V	31.000,-	10.000,-	41.000,-	31.000,-	2.000,-	33.000,-
XI,XII,I-VI	36.000,-	10.000,-	46.000,-	36.000,-	2.000,-	38.000,-
XI,XII +I-VIII	46.000,-	14.000,-	60.000,-	46.000,-	3.000,-	49.000,-

Powrót do Basic'a?

Nie jest naprawdę ważne czy BASIC jest dobry czy zły, z naukowego punktu widzenia, ważne jest czy odpowiada on naszym celom. Jeśli chcesz pisać super szybką grafikę, to Basic okaże się złym wyborem. Jeśli nie miałeś do tej pory do czynienia z programowaniem, to Basic będzie dobrym sposobem na rozpoczęcie nauki.

Nie myśl, że język musi mieć dostęp do grafiki i dźwięku Amigi aby być dobrym - z wielu powodów możesz nie potrzebować tych specyficznych funkcji. Wchodzenie w środowisko Amigi może nie być tak ważne, jeśli np. chcesz tylko ogólnikowo zapoznać się z tematem.

W artykule przedstawiamy skrótowo niektóre jakościowe różnice poszczególnych wersji języka, nie skupiając się na prędkości ich działania. Marginesowe różnice szybkości zwykle nie mają znaczenia i takie przestrzenie, jak grafika i intensywne obliczenia matematyczne, gdzie wzorcowe obliczenia mogą zwykle pomóc, wydają się mniej użytecznym środkiem oceny działania interpretatora i kompilatora, ponieważ rzeczywistą pracę wykonują zwykle procedury z biblioteki Amigi. Wszystko co musi zrobić interpretator lub kompilator, to „wepchnąć" kilka wartości do 68k rejestru procesora i potem przywołać odpowiednią funkcję z biblioteki.

Jest to, jeden z powodów, dla których porównanie niektórych wzorców prędkości kompilator/interpretator, wykazuje małą różnicę.

Microsoft Amiga Basic

Wersja ta dostarczana jest bezpłatnie wraz z komputerem. Główne jej niedogodności to ogólna powolność odtwarzania okna edytora, oraz - oparte na przerwaniach - komunikaty błędów składni.

Większość programistów Basic'u po zrobieniu pierwszych kroków, nie robi już z linii na linię błędów składniowych, a jedyny sposób na problemy z edytorem, to po prostu nie używać go. Często piszę moje programy w Amiga Basic'u używając edytora Lattice LSE i wczytuję je do interpretatora Basic'u Amigi, tylko wtedy gdy są gotowe do końcowego sprawdzenia.

Kompilator Hisft Basic

Jest wiele przesłanek przemawiających za kompilowaniem programów pisanych w Microsoft Basic'u i właśnie kompilator Hisoft doskonale się do tego nadaje. Kompilator jest wyposażony w nowy edytor, w którym pozbyto się powolności oryginalnych propozycji Microsoft. Dostępny jest także rozszerzony pakiet, który wspiera programowanie dla gadgetów, grafiki, dźwięku i plików IFF. Dokumentacja Hisoft'a jest dobra i zawiera przystosowawcze uwagi i dodatki, użyteczne dla bardziej zaawansowanych programistów.

Hisoft kompilator jest przydatny dla programów, które mają być przenoszone z maszyny na maszynę. Powodem tego jest stały wzrost znaczenia Microsoft Basic'u w mikroświecie.

Cursor V1.0

Cursor to kompilator dla programów pisanych w Amiga Basic'u. Dostępny jest jako program Public Domain.

GFA Basic.

Mimo wielu ciepłych słów, którymi możemy obdarzyć oferty oparte na Microsoft Basic'u, prawdą jest także, że istnieje ciągle jeszcze wiele problemów, które można rozwiązać używając tylko procedur biblioteki Amigi. Są one jednak dla wielu programistów Basic'u sporą przeszkodą.

GFA Basic firmy DataMedia z Wielkiej Brytanii, rozwiązał niewątpliwie większość tych trudności, dostarczając wielu rozszerzeń, co czyni pracę programistów łatwiejszą. W niektórych przypadkach pozwala to wykonywać rzeczy, które wydawały się niemożliwe - szczególnie w dziedzinie grafiki i dźwięku. Pakiet Basic'u GFA jest oparty na interpretatorze z ulepszonym kompilatorem GFA.

Takie problemy, jak przemiana plików IFF do Basic'a - przypadkowo ten sam format jak w Microsoft Amiga Basic - są rozwiązywane przez przez GFA Basic w bardzo dogodny sposób. Jedyną wadą jest to, że rozszerzenie GFA uniemożliwiają przenoszenie kodu na inne maszyny.

GFA Basic jest błyskotliwy i jeśli chcesz się „dobrać" do większości obszaru Amigi i znaleźć pomost do późniejszego przejścia w świat języka C i Assemblera to wydaje się, że propozycja Data Media jest interesująca.

Jakikolwiek Basic wybierzesz, jedno jest pewne - powinieneś zapoznać się z podstawowymi rozkazami w Microsoft Amiga Basic'u. Jest on wciąż uważany przez wielu, jako de facto standard i cokolwiek nauczysz się z tego obszaru, będzie ci przydatne w przyszłości przy różnych maszynach.

Opracował H. S.

Na podstawie Amiga Computing 10/91.

Listy, listy

Mam zamiar zakupić Amigę, syntezator i Midi. Proszę o wiadomości na temat programów typu syntezatora najlepiej nadającego się do tego komputera, który będzie dostępny w kraju i w miarę tani, oraz gdzie i za ile można nabyć Midi. Artur Jaroszek.

Można użyć każdy syntezator, który ma wyjście umożliwiające podłączenie Midi, natomiast jeżeli chodzi o typ, to jest to kwestia ceny: od najtańszych poprzez modele średniej klasy po bardzo drogie.

Z programów dostępnych na naszym rynku prawie wszystkie mają obsługę Midi. Jednak polecałbym programy ProTracker lub OctaMED - są bardzo wygodne.

Co do ostatniej części pytania to proszę szukać w sklepach z osprzętem do Amigi lub na giełdach: średnia cena Midi waha się od 300 do 800 tysięcy złotych - oczywiście można dostać taniej jak i drożej - jest to znowu sprawa klasy.

Wasz cykl „Własne demo - Amiga" jest świetny! Mam w związku z tym kilka pytań:
1) Jak wyświetlić obrazki z DeLuxe Painta - nie mogę sobie poradzić z kompresją „byterun"?
2) Jak sprawdzić aktualną ilość dostępnej pamięci CHIP i FAST?
3) Jak zamknąć okno DOS'u i Workbench'a?
4) Jak korzystać z systemowych fontów?
5) Czy modulo w blitterze może być ujemne?
6) Jak włączyć tryb EHB?
Wasz czytelnik.

1) Najlepszą odpowiedzią na pierwsze pytanie będzie krótki program źródłowy, który dekompresuje skompresowany obrazek tworząc z niego występujące po sobie ekrany.

```
; Depack Byte1Run Compressed Body;
I: A0 - Packed Data, A1 - Destination Path
;    D0 - Planes, D1 - Bytes Per Row, D2 - Height
ul_DepackIFF    movem.l    d0-d7/a0-a6,-(sp)
        and.l   #$fff,d1
        and.l   #$fff,d2
        move.l  d1,d3
        mulu    d2,d3
        move.l  a1,a2
        subq.l  #1,d2
        subq.l  #1,d0U
2_Get   move.l  d0,d6
        move.l  a2,a3
U2_NextPlane    move.l     a3,a1
        moveq   #0,d7
U2_TakeNext     moveq      #0,d5
        move.b  (a0)+,d5
        bmi.s   U2_Xero
U2_Copy move.b          (a0)+,(a1)+
        addq.l  #1,d7
        dbf     d5,U2_Copy
U2_Check        cmp.w      d1,d7
        bcs.s   U2_TakeNext
        add.l   d3,a3
        dbf     d6,U2_NextPlane
        add.l   d1,a2
        dbf     d2,U2_Get
        movem l (sp)+,d0-d7/a0-a6
        rts
U2_Xero         move.b     (a0)+,d4
        neg.b   d5
PowielajPetla   move.b     d4,(a1)+
        addq.l  #1,d7
        dbf     d5,PowielajPetla
        bra.s   U2_Check
```

2) Do sprawdzania ilości dostępnej pamięci Chip i Fast służy procedura z exec.library pod nazwą AvailMem(). Aby uzyskać ilość wolnej pamięci podajemy jej w D1 odpowiednie znaczniki tak jak dla AllocMem (patrz artykuł w C64+4), a w D0 otrzymujemy ilość wolnej pamięci.

3) Aby zamknąć okno DOS'u i Workbench'a należy:

```
Exec            =       4
OldOpenLibrary          equu    -$198
ib_ActiveWindow         =       $34
CloseWindow             =       -72
CloseWorkBench          =       -78
        move.l  Exec.w,a6
        lea     IntName(pc),a1
        jsr     OldOpenLibrary(a6)
        move.l  d0,a6
        move.l  ib_ActiveWindow(a6),a0
        jsr     CloseWindow(a6)
        jsr     CloseWorkbench(a6)
IntName dc.b    'intuition.library',0
```

4) Aby korzystać z systemowych fontów należy użyć biblioteki diskfont.library (OpenDiskFont()) oraz korzystać z graphics.library - nie jest to sprawa prosta do wytłumaczenia w tej skromnej rubryce, dlatego postaramy poświęcić jej w najbliższym czasie specjalny artykuł.

5) Właściwie to czemu nie?

6) Aby włączyć tryb Extra Half Bright (dla niewtajemniczonych: 64 kolory) wystarczy włączyć 6 bitplane'ów.

1. *Jaka jest różnica pomiędzy WorkBench'em 1.3.2. a 1.3.3.?*

2. *Gdzie można nabyć nowy system operacyjny Kickstart 2.0 do Amigi 500? Wersję dyskową lub w postaci ROM? Czy na kościach ROM jest on dołączany w postaci Modułu zewnętrznego czy w jakiejś innej? Ile*

kosztuje jego wersja dyskowa a ile ROM?
Adrian S.

1) Jeżeli chodzi o różnice pomiędzy Workbenchami to nie wiem, ale o ile znam życie to jest to różnica tylko pomiędzy numerami.
2) Kickstart 2.0 oryginalnie dostarczany w kościach lub czasami jako bootujący się (dla Amigi 1000) można nabyć u jakiejś firmy wysyłkowej (zachodniej) lub w sklepach na zachodzie. Wersja ROM jest to kość wkładana w podstawkę w miejsce starego układu (czasami są to dwie kości z przełącznikiem - Kickstart 1.3 i 2.0). Co do ceny to trudno mi się jest wypowiedzieć, mogę jedynie stwierdzić, że są to nieduże koszta w stosunku do tego co otrzymujemy. Natomiast jeżeli chodzi o dyskową wersję dla Amigi 500 i 2000 to ze smutkiem muszę stwierdzić, że jest to jedynie piracka robota (chociaż bardzo dobrze wykonana) i nie jest możliwe jej rozprowadzanie.

R*ozkaz DATE czasami zeruje zawartość pliku, któremu chciałem zapisać datę (DATE TO (plik) (data) (czas)). Dlaczego? Oraz skąd mogę dowiedzieć się o zapisie dysku przez Amigę (bloki, sektory, itd.).*
Darek Zakrzewski.

Jeżeli chodzi o przeznaczenie rozkazu DATE to jest ono troszeczkę inne, mianowicie ustawia zegar systemowy, a w składni podanej przez pana jest to skierowanie wszystkich wydruków do danego zbioru a nie ustawienie jego daty. Proponuję napisać DATE TO RAM:XX - zostanie utworzony w RAM-dysku plik o nazwie XX, proszę go odczytać - jest to plik tekstowy zawierający aktualną datę. Natomiast co do drugiej części pytania, to poświęcimy niebawem cały artykuł na ten temat.

1. *Czy Amiga 500 lub 2000 ma wbudowany jakiś język programowania, a jeżeli tak to jaki?*

2. *Czy Amiga ma wbudowany asembler, a jeżeli nie to proszę o podanie nazw najodpowiedniejszych monitorów i asemblerów.*

3. *Czy Amiga 500 jest kompatybilna z Amigą 2000?*

4. *W jaki sposób można odczytywać np. stan klawiatury, wypisywać tekst na ekran, podając tylko adres tekstu w pamięci z poziomu asemblera.*

5. *Czy moglibyście w którymś z przyszłych numerów „C64+4 & Amiga"" wydrukować rozkazy Motorolli razem z liczbami odpowiadającymi tym rozkazom wpisywanym m.in. w liniach DATA.*
Artur Pendzich.

Amiga 500 i Amiga 2000 to praktycznie dwa takie same komputery, więc jeżeli chodzi o stronę programową to są one w stu procentach kompatybilne, natomiast jeżeli chodzi o osprzęt to niestety ale nie. Amiga 500/2000 nie posiada wbudowanego żadnego języka programowania ani żadnego asemblera a najodpowiedniejszymi (przynajmniej dla mnie) asemblerami są MasterSeka 1.80 i AsmOne. MasterSeka jest programem ShareWare i można ją dostać wszędzie natomiast AsmOne jest programem komercjalnym i też można go dostać wszędzie jednak są to pirackie, kradzione wersje tego programu. Natomiast z monitorów dostępnych dla Amigi polecam program D-Mon Pro v3.0 mojego autorstwa (patrz strona druga).
Odczytywania klawiatury można dokonywać na dwa sposoby: z użyciem systemu i bezpośrednio. Przy czym z użyciem systemu jest to dosyć pracochłonne, natomiast bez użycia odbywa się to za pomocą komórki $bfec01. Sprawa ma się podobnie jeżeli chodzi o tekst lecz są to dłuższe procedury.
Nie, brzmi odpowiedź na ostatnie, piąte pytanie, gdyż jest to troszeczkę bez sensu, a poza tym zajęło by to chyba cały numer.

W *jaki sposób można zajrzeć do gier na Amidze? Jak można przerwać grę tak aby nie uległa skasowaniu?*
Marcin Bohdziul.

Aby zajrzeć do gry w czasie gdy jest ona uruchomiona istnieje tylko jedna metoda - przerwać ją za pomocą opcji freeze w cartridge'u (np. Action Replay). Można ją wtedy dowolnie modyfikować i powrócić do niej. Istnieją jeszcze programy pozwalające na kontrolowanie niektórych gier, ale tylko niewielki procent gier (a zwłaszcza nowych) daje się kontrolować. Jeżeli chodzi natomist o zaglądanie do gier przed uruchomieniem to wystarczy dowolny monitor języka maszynowego.

Marcin „Duddie" Dudar

KURS JĘZYKA C (cz.3)

Poprzedni odcinek zawierał: tablice, instrukcje for, while, do...while, if, operatory inkrementacji i dekrementacji.

Język C zawiera szereg typów zmiennych. Warto się jednak zastanowić nad tym, co będzie, gdy dodamy do siebie zmienną typu **int** do zmiennej typu **double**. Jaki będzie typ wyniku takiej operacji?

Przekształcenia typów następują według ściśle określonych zasad. Generalny sposób postępowania polega na przekształceniu danych do jednego wspólnego typu. Bezsensowne działania, jak np. próba indeksowania tablicy zmienną typu **float** nie są realizowane.

Przejdźmy do szczegółowego opisu. W języku C nie istnieje praktycznie żadna różnica pomiędzy zmiennymi **typu int i char**, które mogą być przemieszane dowolnie. Tak więc poniższy program jest poprawny:

```
main()
{
 int kod;              /* Zmienna typu char zawierajaca
                        kod ASCII */
 kod = 53;
 printf( "\nKod = %d, liczba = %d\n", kod, kod-'0' );
}
```

Warto zwrócić tu uwagę na wyrażenie: kod-'0'. Otóż od danej typu **int** odejmujemy wartość stałej tekstowej (stałe te umieszcza się pomiędzy znakami apostrofów). Ponieważ zmienna kod ma wartość ASCII odpowiadająca znakowi '5', więc odjęcie wartości odpowiadającej kodowi ASCII znaku '0' da w efekcie wartość 5.

Konwersja z typu **char na int** niesie z sobą pewne niebezpieczeństwo. Otóż pisząc programy na różne maszyny należy zwrócić uwagę, czy powstała w wyniku konwersji zmienna **int** jest ze znakiem czy bez.

Zajmijmy się pozostałymi typami danych. Poniżej przytoczono kilka sposobów konwersji, które są odzwierciedleniem generalnej zasady polegającej na przenoszeniu danej mniejszego typu (w sensie rozmiaru) do typu większego:

1. char jest przeksztalcany na int.
2. short jest przeksztalcany na int.
3. float jest przeksztalcany na double.

Jeżeli wykonujemy operacje na dwóch danych, to najpierw konwersja odbywa się według powyższego wzorca. Następnie jeśli jakikolwiek z argumentów jest typu **double**, to drugi także zostanie przekształcony do tego typu. Jeśli typem jednego z argumentów jest **long**, to drugi także zostanie przekształcony do typu **long**. Podobnie jeśli któryś z argumentów jest typu **unsigned**, to drugi także zostanie przekształcony do typu **unsigned**.

Oto kilka dodatkowych informacji o własnościach niektórych przekształceń. Zamiana **float** na **int** obcina część ułamkową, **double** na **float** jest przekształcany przez zaokrąglenie. Typ **long** przekształcany jest na krótszy (np. **short** lub **char**) przez obcięcie najbardziej wystających bitów.

To tyle ogólnych zasad konwersji. W praktyce pamiętanie powyższych informacji nie jest konieczne - wystarczy pisząc jakieś wyrażenie po prostu zastanowić się, czy ma ono sens.

Programy w języku C pisze się w sposób strukturalny, tj. poprzez tworzenie funkcji. Każdy program w C jest funkcją. Używana przez nas w każdym programie nazwa **main()** jest również niczym innym, jak nazwą funkcji. Z wyjątkiem słowa **main** nazwa funkcji może być praktycznie dowolna. Oto przykład programu korzystającego z dwóch funkcji: głównej - **main()** - od której rozpoczyna się działanie programu i funkcji o nazwie **Wydruk()**:

```
void Wydruk()
{
 printf( "\nOto efekt dzialania funkcji Wydruk()\n");
}

main()
{
 Wydruk();
}
```

Ogólnie każdą definicję funkcji umieszczamy przed funkcją **main()**. W przypadku umieszczenia definicji funkcji za main należy na początku programu umieścić deklarację funkcji:

```
void Wydruk();

main()
{
 Wydruk();
}

void Wydruk()
{
 printf( "\nOto efekt dzialania funkcji Wydruk()\n");
}
```

Tajemnicze słowo **void** przed nazwą funkcji mówi o tym, że funkcja nie zwraca żadnej wartości. Jeśli w wyniku działania funkcji miałaby zostać zwrócona jakaś dana, należy przed nazwą funkcji umieścić jej typ:

```
short Znak()
{
 return(1);
}

main()
{
 printf( "\nFunkcja znak zwrocila wartosc: %d\n", Znak());
}
```

W tym przypadku w definicji funkcji umieszczono informację o tym, że zwracana przez funkcje **Znak()** wartość jest typu **short**. Występujące nowe polecenie **return** ma dwojaki format: może wystepować jako rozkaz bezwzględnego zakończenia działania danej funkcji bez zwracania danej (**return;**) lub ze zwróceniem danej (**return(dana);**).

Funkcja z reguły posiada jakieś parametry wywołania. Umieszczane są one w nawiasie za nazwą funkcji:

```
void RozneWydruki ( short liczba )
{
 if (liczba)
   printf( "\nPodano parametr rozny od zera!\n");
 else
   printf( "\nPadano parametr rowny zeru!\n");
}

main()
{
 RozneWydruki(1);
 RozneWydruki(0);
}
```

Funkcja **RóżneWydruki()** ma jeden parametr rozmiaru **short** o nazwie **liczba**. Jeśli parametr ten jest różny od zera następuje wydrukowanie jednego tekstu, jeśli równy zeru - innego. Wartość parametru jest przekazywana funkcji w momencie jej wywołania.

Istotne jest zwrócenie uwagi na to, że funkcja jako parametr otrzymuje wartość zmiennej, a nie tę zmienną. Oznacza to, że modyfikacja przekazanej zmiennej dokonana wewnątrz funkcji nie będzie faktycznie modyfikacją przekazywanej danej. Najlepiej zobaczmy to na przykładzie:

```
void Proba ( short dana )
{
 printf( "\n W wywolywanej funkcji pierwotna wartosc
         wynosi: %d\n", dana );
 dana++;
 printf( "\n W wywolywanej funkcji wtorna wartosc
         wynosi: %d\n", dana );
}

main()
{
 int dana;
 dana = 5;
 printf( "\n W glownej funkcji pierwotna wartosc wynosi:
         %d\n", dana );
 Proba( dana );
 printf( "\n W glownej funkcji wtorna wartosc wynosi:
         %d\n", dana );
}
```

Jak widać, i przed i po wywołaniu funkcji wartość zmiennej dana jest równa 5. Bierze się to stąd, że w momencie wywoływania funkcji pojawia się zmienna, której nadaje się wartość równą wartości danej będącej parametrem wywołania. Zmienna ta jest nową zmienną i w naszym przykładzie tylko przypadkowo jej nazwa zbiega się z nazwą danej będącej parametrem. O tym, że jednak można zmodyfikować przekazywane dane z wnętrza funkcji będzie mowa w następnych częściach kursu.

Istnieją dwa style zapisywania parametrów funkcji. Oto styl pierwszy:

```
void funkcja ( int parametr1, short parametr2 )
```

a to drugi:

```
void funkcja ( parametr1, parametr2 )
 int parametr1;
 short parametr2;
```

Osobiście wolę styl pierwszy i ten też będzie przeze mnie używany. Dzięki niemu definicja funkcji jest wyraźnie widoczna w tekscie programu, poza tym parametry mają określany typ bezpośrednio w momencie definiowania funkcji.

Każda funkcja może operować na zmiennych. Zmienne te mogą być deklarowane w ciele funkcji, np.:

```
void Liczby()
{
 int loop;
 for ( loop=0; loop; loop++ )
   printf( "\nKolejna liczba jest rowna %d", loop );
}

main()
{
 Liczby();
}
```

Zmienna **loop** typu **int** została w standardowy sposób zadeklarowana w ciele funkcji. Jest ona używana do wyprowadzenia dziesięciu wartości liczbowych i jest widoczna jedynie wewnątrz funkcji **Liczby()**. Oznacza to, że nazwa **loop** może wystąpić w odniesieniu do więcej niż jednej zmiennej i nie spowoduje to żadnych kolizji, o ile zmienne te są zwiazane z różnymi funkcjami. Np.

```
void LiczbyRosnaco()
{
 int loop;
 for ( loop=0; loop; loop++ )
   printf( "\nKolejna liczba jest rowna %d", loop );
}

void LiczbyMalejaco()
{
 int loop;
 for ( loop=9; loop0; loop-- )
   printf( "\nKolejna liczba jest rowna %d", loop );
}

main()
{
 int loop;
 for ( loop=0; loop; loop++ )
  {
   LiczbyRosnaco();
   LiczbyMalejaco();
  }
}
```

Jak się zachowują zmienne w funkcjach, jakie występują klasy zmiennych to z kolei temat na następną część kursu.

Jarosław Chrostowski

Flight of the Intruder czyli symulujemy w powietrzu

I oto w końcu jest! Eleganckie kartonowe pudełko opakowane w termokurczliwą folię. Ale to dopiero początek. W środku oprócz dwóch dyskietek znajdujemy ponad 200 stronicową instrukcję, mapę taktyczną z obszarem pola walki oraz książkę traktującą o wojnie powietrznej nad Wietnamem. I to wszystko po to, aby nas wprowadzić w realia pilotowania dwóch śmiercionośnych maszyn jakimi są: A-6 Intruder oraz F-4 Phanton. Pełni niecierpliwości ładujemy grę i wkraczamy w świat, w którym można latać nawet na cegle, jeżeli tylko mamy odpowiednio mocne silniki.

Na wstępie podajemy nasze imię oraz znak rozpoznawczy dla naszego samolotu, teraz mamy do wyboru dwa rodzaje scenariuszy (dla Phantona i dla Intrudera). Po wciśnięciu klawisza F10 ukazuje się menu, w którym ustawiamy wszystkie parametry lotu. Wybieramy misję i już możemy startować z lotniskowca. Przekrój misji (od prostego bombardowania statków po kompleksowe naloty na całe miasta) zadowoli największego maniaka symulacji.

Istnieje również możliwość zaprojektowania własnej misji, gdzie celami będą autentyczne cele na terenie Wietnamu. W bardziej skomplikowanych akcjach bierze udział większa ilość maszyn, przy czym w odróżnieniu do innych programów tu możemy sterować wieloma samolotami w czasie jednej misji. Istnieje również możliwość gry przez modem, gdzie obaj gracze albo walczą razem po jednej stronie, albo przeciwko sobie. W czasie lotu możesz oglądać swój samolot z różnych ujęć tak z wewnątrz, jak i z zewnątrz. Istnieje opcja, w której kamera umieszczona jest wewnątrz pocisku co pozwala obserwować skuteczność ataku. Możemy nagrać część akcji na kasetę video, którą przegląda się po jej zakończeniu (istnieje też opcja robienia zdjęć). Ogólnie można powiedzieć, że program jest bardzo starannie dopracowany mimo że miejscami „wektorówka" zwalnia (duża ilość obiektów na ekranie).

Smacznym kąskiem dla wszelkich maniaków jest manual, gdzie poza danymi technicznymi obu samolotów i opisem miejsca i czasu akcji podano szczegółowe dane dotyczące rodzajów uzbrojenia, techniki walki powietrznej, sposobów przeprowadzania nalotów, techniki bombardowania, oraz lądowania na lotniskowcu (czytaj: lądowania słoniem na znaczku pocztowym).

To wszystko uzyskujemy po zakupieniu licencjonowanej kopii, a nie w przypadku skopiowania „połamanej" gry u kolegi. Raz jeszcze potwierdza się prawda, że lepiej kupować oryginały, ale o tym już pisaliśmy w jednym z poprzednich numerów. Niemniej jednak wszystkim życzę owocnych godzin spędzonych przy tym symulatorze, który całkowicie zasłużenie zajmuje pierwsze miejsce na liście symulatorów lotu.

Robert „Mr.Raf" Turliński

Kolejna cegła w murze czyli Datel w natarciu

Któż nie zna firmy Datel, która od lat produkuje urządzenia peryferyjne do C - 64 i Amigi. To właśnie m.in. ona wystawiała swoje najnowsze osiągnięcia na ostatnich targach w Niemczech. Stamtąd dzięki Panu Piotrowi Drapichowi, oficjalnemu dystrybutorowi urządzeń firmy Datel otrzymaliśmy do testu kolejną "cegłę" czyli ... **ActionReplayMKIII.**

Zewnętrznie moduł nie zmienił się wcale (trochę inne kolorki), więc bez zbędnych oględzin „wbiłem" go w złącze. Po włączeniu zasilania pojawia się strona tytułowa (nowa grafika, całkiem zbędna moim zdaniem) i nasza „cegła" sygnalizuje gotowość do pracy zieloną diodą. Wciskam dobrze znany czerwony przycisk i zaczynam test.

Pierwszą rzeczą, która rzuca się w oczy to fakt, że w końcu mamy możliwość używania małych liter. Jak zwykle po wciśnięciu przycisku „Help" otrzymujemy spis dostępnych komend (oczywiście mocno poszerzony). Z plusów można wymienić poprawiony „zamrażacz" (freeze) programów, który po ponownym wejściu do programu uruchamia stację dysków (sprawdzone na grze SWIV). Można też zatrzymać dowolny program, czego nie da się powiedzieć o poprzedniej wersji (sprawdzone na nowym Snooker'ze). Pojawiła się opcja kontroli bootblock'u jako osobny gadżet co umożliwia nam wyłączenie czyszczenia pamięci i poszukiwania wirusów w pamięci. Dzięki temu możemy spokojnie pracować z RAD, RRD lub VD0 bez obawy o stratę danych w pamięci, a jednocześnie moduł sprawdza wszystkie wkładane dyskietki, czy nie zawierają wirusów.

Maniacy gier mają do dyspozycji menu wyszukiwania nieśmiertelności poszerzone o opcję głębokiego przeszukiwania pamięci oraz coś co firma określa jako „Ultimate Joystick Handler". Pozwala on definiować dodatkowe klawisze dostępne w grze za pomocą kombinacji fire'a i położenia joystick'a. Jednak tak jak nie ma produktu doskonałego, tak i w tym przepadku mamy kilka mankamentów. Po pierwsze moduł nie jest kompatybilny z poprzednikiem jeśli chodzi o zbiory zrzucane opcją „sr". Jeśli chcemy wczytać taki zbiór musimy wychodzić do dos'u i używać starej procedury „aload". Po drugie nie działa opcja „tfd" (search and remove decrement opcodes), albo działa zbyt dokładnie (czekałem 30 minut i nic się nie działo więc niewiem). Dotychczas nie zauważyłem innych błędów.

Na zakończenie należy dodać, że otrzymany od pana Piotra moduł jest pierwszym jaki pojawił się w Polsce i dlatego test przeprowadzałem bez firmowej instrukcji. Jeśli tylko takowa napłynie do nas z firmy „InterComp" to natychmiast zamieścimy bardziej szczegółowy test AR MKII.

Robert „Mr.Raf" Turliński

Urządzenie do testowania udostępniła firma InterComp sp. z o.o., mieszcząca się w Warszawie przy ul. Karowej 18a/20 (tel. 022-266607).

Piątek, 15 listopada, godz. 17:00. Mimo wielu niepowodzeń, wykruszenia się większości składu, etc. zdecydowaliśmy się opuścić nasze kochane miasto i udać na C-Party do Gdyni. Przygotowania polegały głównie na wrzuceniu do torby około 30 dysków - niekoniecznie pustych, zapakowaniu taśmy szczęścia, szczoteczki do ząbków szorowania oraz innych, niezbędnych akcesoriów (czytaj: pieniądze).

Sobota, 16 listopada, godz. 2:00. Po „odpoczynku" w jednej z szczecińskich dyskotek udajemy się na dworzec. Jak wszystkim wiadomo Polska jest krajem gdzie wszystko robi się troszeczkę inaczej, ale żeby aż tak... Wróćmy więc do początku. Po dotarciu na dworzec w Szczecinie, odszukaliśmy jedyną czynną kasę (pierwszy sukces, a może porażka...) i po zadaniu pytania o rezerwację miejsc na Express, godz. 6:20 usłyszeliśmy odpowiedź, że owszem możemy sobie wykupić bilety z miejscówkami, ale na dwie godziny przed odjazdem pociągu... BOMBA!!! Z tego co się do tej pory orientowałem to zawsze można było zarezerwować miejsca troszeczkę wcześniej i dlatego nazwano to rezerwacją miejsc. Ale cóż czasy się zmieniają i obyczaje również. Nic nam to nie przeszkadzało w kontynuowaniu zabawy więc poczekaliśmy chwilkę, nabyliśmy bilety i udaliśmy się na peron...

Sobota, 16 listopada, 1991r. godz. 6:20. Pociąg. HURRA!!!

Sobota, 16 listopada, godz. 10:40. Gdynia Główna Osobowa nie przywitała nas fajerwerkami, na planie miasta nie znaleźliśmy Domu Stoczniowca, w którym Copy Party miało mieć miejsce (zgubiłem gdzieś zaproszenie, na którym była mała mapa poglądowa jak na party trafić).

W końcu z pomocą osób trzecich znaleźliśmy się pod Domem Stoczniowca i... i znowu żadnej informacji. Wchodzimy ostro... a tu jeszcze ostrzej wyskakują do nas z pytaniem: Na Copy Party??? Na co my: Noooo!!!!

Poprowadzili nas przez zdezelowaną salę konferencyjną, parę korytarzy (nie zawiązywali nam oczu no bo po co? I tak byśmy nie trafili z powrotem). Kiedy wreszcie dotarliśmy na miejsce zabrano nam po czterdzieści tysięcy złotych polskich od łebka, zrobiono identyfikatory i dopiero wtedy dopuszczono, jako pełnoprawnych uczestników C-Party (mieliśmy pewne kłopoty jeżeli chodzi o wpisanie nazwy grupy na identyfikator, ale sobie szybciutko z tym poradziliśmy wpisując wysłużony Quartet Inc.).

Po wejściu na Party stwierdziliśmy, że coś się zmieniło!!! Tylko co.... aaaa... już wiem... rzadko kto używał X-Copy (chociaż nie znaczy to, że go nie używano). Scena przeżyła gwałtowny rozwój. Pojawiło się wiele grup (i to wcale nie pseudo grup), które produkują całkiem niezłe rzeczy. Po raz pierwszy udało się przeprowadzić Competition czyli konkurs na najlepsze produkty, tj. muzykę, grafikę oraz demo, ale o tym za chwilę.

Party nawiedziło wiele osób i o niektórych z nich należy wspomnieć: grupy Joker i Luzers były dosyć licznie reprezentowane (jako organizatorzy musieli się pojawić), z grupy Katharsis przybyło kilku członków a mianowicie: Raf, Mac i Jerry, jednoosobowa sekcja polska grupy ACME - Mr.Root (na party pojawił się jego nowy przydomek: Mr.SMRoot of LAcme) a także Dr.Piotr (Docent), który zajął się na poważnie interesami i oferował trochę sprzętu (Mr.Raf pożyczył go Action Replay'a III) oraz oryginalny polski program - Virus Expert II (którego kopię otrzymałem i którego test przeczytacie niebawem w „C64+4").

Party odwiedziło też wiele innych grup takich jak: H.O.T., Addonic, Old Bulls, Zack Team i inne.

Ważnym atrybutem imprezy był barek, w którym można było nabyć napoje gaszące pragnienie: 7UP, Pepsi, Kawa, Herbata, i inne, a także coś do szamania: BIGOS, FASOLKA PO BRETOŃSKU i PARÓWY. A propos szamania to Raf z grupy Katharsis zajmował się efektywnym oczyszczaniem barku z bigosu, a robił to stale, ciągle i namiętnie, w tempie zaiste diabelskim.

Tak nam miło upływał dzień, że nie zauważyliśmy jak nadszedł czas konkursów. Najpierw zebrało się gremium i wśród krzyku, gwaru, nisko przelatujących krzeseł i Amig dało się słyszeć z początku niepewne, następnie troszkę głośniejsze a na końcu świdrujące narząd słuch muzyczki - przypomniały mi się od razu wszystkie moje kompozycje i stwierdziłem, że nie jestem wcale złym muzykiem (he, he...). Okazało się, że najlepszym muzykiem w Polsce jest Mr.Root i co by nie mówić chyba gremium jurorskie miało rację chociaż muzyczki wystawione przez niego do competition nie należały do jego najlepszych kompozycji. Potem był konkurs na najlepszego grafika, który niestety przegapiłem, oraz konkurs na najlepsze demo. I tu niespodzianka: okazało się, że nasze rodzime grupy potrafią dać z siebie bardzo dużo a nawet więcej. Do faworytów należały dema grup: H.O.T. i Joker. Demko HOT'ów to popis kodowania i to kodowania na bardzo wysokim poziomie z wieloma wspaniałymi pomysłami i ich genialnym wykonaniem natomiast demo Jokerów to przykład idealnego dopracowania i staranności. Wspaniała animacja RYSia znajdująca się na początku dema „zabiła" niejednego. Reszta była także wspaniała. Uważam, że te dwa dema powinny dostać ex equo pierwsze miejsce!!!

Tak zakończyło się competition, a wraz z nim dzień pierwszy Copy Party.

	demo	muzyka	grafika
1	THE RETURN-JOKER	MR.ROOT /ACME	ANIMAL /ACTION DIRECT
2	POWER - HOT	XTD /HAZE	SEQ /ZACK TEAM
3	DEFENE - LUZERS	PIC - SAINTLO UP	DAN/G-FORCE

Wyniki Competition

Sobota, 16 listopada, godz. 22:00. Po zakończeniu wszelkich atrakcji udaliśmy się na czwarte piętro Domu Stoczniowca, gdzie mieliśmy zarezerwowane pokoje po tanich kosztach - trzydziestu pięciu tysięcy za głowę (albo ciało potrzebujące łóżka)).

Niedziela, 17 listopada, godz. 10:00. Drugi dzień C-Party zaskoczył nas dźwiękiem muzyki z dema Jokerów (bo skoro wygrało party to jest najlepsze i należy je puszczać wszędzie, aż do znudzenia!). Kopiowanie nasiliło się w porównaniu do dnia wczorajszego znacznie i opanowało ok. 90% sali (w porywach do 100%). Jakiś młodociany lamer poprosił mnie, abym mu zrobił scroll'a za co będę mógł sobie wybrać jakiś opis. Oczęta mi się zaiskrzyły bo zobaczyłem opis do Imagine'a (nie powiem żebym potrzebował jej aż tak bardzo, ale jeszcze kilku rzeczy w Imagine nie wiem). Po zmontowaniu scroll'a (nawet fonty zrobiłem) małolat mi powiedział żebym sobie wziął jakiś opis za wyjątkiem Imagine...

Grupa HOT z pewną inną grupą (niepamiętam czy był to ZACK TEAM czy OLD BULLS) stwierdziły, że się rozwiązują i razem tworzą nową grupę pod nie wiadomo jaką nazwą.

Po skonsumowaniu kilku (kilkudziesięciu) porcji parówek udaliśmy się na dworzec i odbyliśmy podróż do Szczecina.

Niedziela, 17listopada, godz. 16:40 - niedziela, 17 listopada, godz. 23:00. Powrót do Szczecina odbywał się całkiem przyjemnie i zakończył się podziałem łupów. Przywieźliśmy kilka gierek, nowy emulator ST pod nazwą Amtari, nowego DiskMastera v2.0, nowy program muzyczny OctaMED i wiele innych drobnych programików.

Podsumowanie:C-Party zorganizowane przez grupy Joker i Luzers było wspaniałą imprezą, na której można było poznać wiele nowych osób oraz całkiem nieźle się bawić. Oby więcej takich.

Marcin „Duddie" Dudar

Secret of the Monkey Island

Po chwilowej przerwie spowodowanej obiektywnymi trudnościami jestem znowu z wami Maniacy gier adventure, by służyć wam pomocą w następnej przygodzie. Na wstępie proponuję nie zaczynać jej w poniedziałek, bo możemy popaść w konflikt z rodzicami i systemem edukacji. Spokojnie siadamy w piątek wieczorem, by w sobotę rano (lub później) zakończyć sukcesem nasze wysiłki!

Meele Island to jedyne miejsce gdzie w dzisiejszych czasach istnieją piraci i gdzie można w krótkim czasie stać się jednym z nich. Nie pytaj mnie, jak się tam znalazłeś, bo tego nie wie nikt. Jeśli już pogadałeś ze ślepym jak nietoperz strażnikiem, to udaj się do jedynej gospody w mieście. Od czasu, gdy kapitan Le Chuck przepędził piratów z morza siedzą oni w „Scumm Bar", spędzając czas na gadaniu bzdur i piciu grogu. Pogadaj z nimi całkiem serio, a dowiesz się jak można osiągnąć wymarzony cel.

Zanim opuścisz tą przytulną norę proponuję byś zwiedził kuchnię i zabrał wszystko co można poruszyć (no nie tak dosłownie, mewę możesz zostawić w spokoju) oraz popróbował swoich sił w gotowaniu.

Teraz udajemy się na ulice miasta by zorietować się w sytuacji. Zachowuj się, jak na pirata przystało i rozmawiaj z wszystkimi, a może dowiesz się paru interesujących rzeczy. Oczywiście, jeśli napotkasz coś co może zmienić właściciela to nie miej skrupułów. Niestety tak to już bywa, że na tym świecie rządzi szmal, więc trzeba go jakoś zdobyć.

Jedynym sensownym (tak???) miejscem, gdzie możesz się udać jest cyrkowy namiot. Wloscy bracia mają trudny problem do rozwiązania i poszukują królika doświadczalnego (mięso armatnie to popłatna praca). Teraz bogatsi o parę groszy udajemy się na zakupy potrzebnego sprzętu (na zakupy to do sklepu, nie do speluny na grog!). Czas na rozpoczęcie pierwszej próby. Najpierw jednak trzeba nauczyć się władać mieczem. Jeśli bardzo tego pragniesz to kapitan Smirk przyjmie cię jako ucznia i przeszkoli w trudnej sztuce walki. To nie wszystko!

Teraz, jak na pirata przystało musisz się nauczyć przeklinać (no ładnie). Zaczepiaj każdego i śmiało walcz, a opłaci się parę razy dostać w skórę. Jeśli mówią ci, że możesz już walczyć z samym mistrzem to nie ociągaj się tylko pędź do lasu (jak się boisz iść bez mapy, to ją kup). Nie zawadzi zebrać trochę kwiatków zanim dotrzesz do mistrza (tylko nie mów, że nie możesz go znaleźć). Tu stawiamy sprawę jasno i po krótkiej wymianie zdań Carla przyznaje ci rację co do twoich umiejętności.

Teraz część druga (nie zapomnij pochwalić się piratom). Przed udaniem się do twierdzy pogadaj z więźniem, a dowiesz się ciekawych rzeczy. Podłe pudle-piranie potraktuj perfidnie przyprawioną potrawą (w skrócie 7XP). W środku wal śmiało do pokoju i spokojnie poczekaj kilka chwil. Potrzebny ci pilnik, a takie o ile mi wiadomo występują w ciastach (podpowiedź: susel przypomina szczura). Jeśli ktoś cię grzecznie zapyta po co ci figurka, którą właśnie ukradłeś to nie kręć tylko mów prawdę i nie wysilaj się przy rozmowie z panią gubernator, bo i tak wyjdziesz na głupka.

Po tych wszystkich gorących chwilach zaproponuję ci orzeźwiającą kąpiel z balastem na co nie masz wpływu. Na dnie zostaw wszystkie ostre przedmioty, bo sobie przy okazji coś utniesz (10 minut to kupa czasu do namysłu, jak się nie utopić). Teraz już tylko poszukiwanie skarbu i zostaniesz piratem, a przy okazji mężem pani gubernator. Skarby zakopywane są zawsze w lesie (nie w kasie pancernej!), a drogę do nich wskazuje stara mapa (to co nazywa się „mapa" jest mapą). Teraz tylko trochę kopania i już jesteś piratem.

Nie dane ci jest jednak połączyć się z kobietą twego życia, gdyż ta została porwana przez podstępnego Le Chuck'a. Nie leń się tylko zaraz ruszaj jej na pomoc. Najpierw skompletuj załogę (odwiedź dom na wyspie) i kup statek. Jeśli masz zbyt mało pieniążków to pogadaj z właścicielem sklepu na ten temat. Gdy już masz fundusze, to w stoczni nie rozglądaj się tylko kup statek z odpowiednią „przeszłością" (jak za drogi to się trochę potarguj). Teraz razem z załogą wyruszamy na poszukiwanie tajemniczej „Monkey Island".

By tam dotrzeć, zdany na własne siły, najpierw przeszukaj cały statek i nie bój się o pojemność swoich kieszeni. Cały problem polega na ugotowaniu zupki, która „dopchnie" cię do celu podróży (najpierw przepis, bo bez niego ani rusz). W jej pichceniu wykaż się domyślnością, by po paru dniach, jak z armaty wylądować na wyspie.

Tutaj duża garść wskazówek co do dalszego postępowania. Na początku przeszukać każdy kawałek (dostępny) wyspy. Pirat bez prochu, to jak samochód bez paliwa. Łódka swoją budową przypomina statek i służy do pływania, natomiast wiosła nie przypominają swoją budową nic, ale bez nich ani rusz. Jeden kawałek liny jest dłuższy od dwóch, a drewno wypływa na powierzchnię wody. Krzemień + żelazo = iskra + proch = bum! Ważny przyrząd służy do miotania kamieni co wynika z samej jego definicji (Pierwszy strzał najważniejszy). Małpy żywią się bananami tylko w dużej ilości i bardzo lubią małpować. Kanibale nie są tacy straszni, lubią prezenty i mają dziurawą podłogi. Herman Zgniloząb chce swój zbieracz (nie mylić ze zwieraczem) bananów. Po labiryntach najlepiej chodzi się z przewodnikiem, bo co dwie głowy to nie jedna. Książka jest dobra na wszystko (to nie moje, ale też fajne). Oprócz czapek niewidek występują również niewidome naszyjniki. Pióra łaskoczą, grog upija, magnes przyciąga, a tłuszcz smaruje (nie wspominając o narzędziach, które same nie robią nic, ale jak je poruszać to ho, ho!).

Jeśli zrozumiałeś o co mi chodzi, to powinieneś być szczęśliwym posiadaczem najsilniejszego środka przeciw duchom jakim jest piwo korzenne (nie spożywać bez uprzedniej konsultacji z lekarzem). Nikt nie jest w stanie oprzeć się takim argumentom, więc śmiało idź do celu wyprawy i nie trać zimnej krwi bez względu na to co się wydarzy.

Na zakończenie pragnę pocieszyć tych, którzy polubili naszego bohatera i smutno jest im się z nim rozstawać. Otóż na wakacje roku 1992 ukaże się druga część naszej fascynującej przygody z jeszcze wspanialszą grafiką, jeszcze śmieszniejszymi tekstami i oby nie z jeszcze większą ilością dysków. Póki jeszcze mogę, pragnę wyrazić podziękowanie Samborowi Kuźmie (druga połówka brygady operacyjno-uderzeniowej) za silną wolę i wiarę w inteligencję zachodnich programistów, oraz Grzegorzowi Mikule za spostrzegawczość (chyba nie pokręciłem pojęć?). Bez tych osób nie było by tego artykułu, a jego autor nie omijałby każdego komputera przez tydzień.

A teraz: Abra, Cadabra, Cobra, Macabra, Przerywam, Łączę i Artykuł Kończę (no kto zgasił światło!!??).

Robert „Mr.Raf" Turliński

ARP LIBRARY (CZ.2)

I oto druga z kolei część opisu biblioteki Arp. W tym odcinku kolejne procedury Arp.

ASyncRun - uruchomienie asynchroniczne

ASyncRun („Nazwa", „Argumenty", PCB)

a0 a1 a2

Utworzenie nowego zadania, na którym zostanie uruchomiony program o podanej nazwie. Wiele typów zadań może być utworzonych włączając styl zadań „CLI". Funkcja ta ma poprawione błędy występujące w funkcji Execute z dos.library. Jeżeli nie zostanie zwrócona wartość ujemna należy się spodziewać, że zadanie zostało przyłączone do systemu i uruchomione. Po zakończeniu danego zadania wszystkie obszary zajmowane przez program będą zwolnione (nie dotyczy to obszarów zaallokowanych przez wykonywany program, np. obszary na dane).

Wejście:

Nazwa - nazwa programu do uruchomienia. Jeżeli nie wpiszemy wartości do komórki pcb_LoadedCode (w strukturze Process Control Block - PCB), to nazwa musi być pełną nazwą programu do załadowania bądź programu rezydentnego. Jeżeli użyjemy już wczytanego programu (pcb_LoadedCode), to nazwa nie musi odwoływać się do jakiegokolwiek programu.

Argumenty - tutaj znajdują się parametry dla uruchamianego programu. Wartość ta może być zerowa ale długość linii argumentów nie może przekroczyć 256 bajtów. Należy zwrócić uwagę, że niektóre programy mogą wykazywać błędy jeżeli nie podamy im argumentów dlatego w skrajnych przypadkach należy linię argumentów zbudować ze znaku końca linii ($a ('/n')) oraz z zera.

PCB - Process Control Block.

Struktura o następującej budowie:

```
STRUCTURE      ProcessControlBlock,0
ULONG              pcb_StackSize
        BYTE       pcb_Pri
        BYTE       pcb_Control
        APTR       pcb_TrapCode
        ULONG      pcb_Input
        ULONG      pcb_Output
        UNION      pcb_Console,4
             UMEMB   pcb_Splatfile
             UMEMB   pcb_ConName
        ULONG      pcb_LoadedCode
        CPTR     pcb_LastGasp
        CPTR     pcb_WBProcess
```

Użytkownik musi koniecznie ustawić komórki pcb_StackSize oraz pcb_Pri. Jeżeli nie chcemy używać pozostałych komórek powinniśmy je wyczyścić.

pcb_StackSize - rozmiar stosu, który będzie przydzielony dla tego zadania. Powinien być wielokrotnością liczby cztery. Jeżeli program uruchamia się wraz z ResidentProgramTag to rozmiar stosu będzie zasąpiony tą wartością.

pcb_Pri - priorytet z jakim ma być uruchomione zadanie.

pcb_TrapCode - adres procedury obsługi przerwań wywołanych błędami. Jeżeli nie wpiszemy żadnej wartości to zostanie tam automatycznie wpisana wartość specjalnego TrapCode dla DOS'u. Powinno się tego używać bardzo uważnie i świadomie.

pcb_Input - standardowy zbiór wejściowy. Zostanie zamknięty przy opuszczeniu procesu jeżeli nie zostanie ustawiony bit PRB_SAVEIO.

pcb_Output - standardowy zbiór wyjściowy. Patrz pcb_Input.

pcb_LoadedCode - jest to wartość zwracana przez LoadSeg lub LoadPrg (Segment). Jeżeli jest ona ustawion to procedura ASyncRun nie będzie wykonywała procedury LoadSeg („Nazwa") oraz nie będzie zwalniała pamięci przy opuszczaniu zadania. Znaczenie tej zmiennej jest uzalcżnione od bitu PRB_CODE.

pcb_LastGasp - jeżeli chcemy uzyskać informację o zakończeniu danego procesu to powinniśmy zostawić tutaj adres struktury ZombieMsg. Struktura ZombieMsg MUSI mieć własny ReplyPort.

```
STRUCTURE     ZombieMsg,MN_SIZE
        ULONG     zm_TaskNum
        ULONG     zm_ReturnCode
        ULONG     zm_Result2`
        STRUCT    zm_ExitTime,ds_SIZEOF
        ULONG     zm_UserInfo
```

pcb_WBProcess - używane tylko w przypadku gdy ustawimy bit PRB_NOCLI oraz gdy zadanie zostanie uruchomione właściwie to w tej komórce znajdzie się port komunikacji z zadaniem.

pcb_Console - zmienna ta składa się z dwóch zmiennych składowych: pcb_Splatfile i pcb_ConName gdzie pcb_Splatfile - zbiór do otworzenia przez procedurę Open, do którego będą wysyłane komunikaty przez program. Jeżeli program jest uruchamiany z CLI wtedy nie mamy żadnego problemu natomiast w przypadku uruchamiania pod Workbenchem musimy otworzyć odpowiednie okno na komunikaty.

pcb_Flags - ta zmienna zawiera różne znaczniki kontrolne i tak:

PRB_SAVEIO (bit 0) - wymusza na zadaniu pozostawienie otwartych kanałów Wejścia/Wyjścia w momencie gdy zostanie ukończone to zadanie. Możemy dzięki temu użyć tych kanałów powtórnie.

PRB_CLOSESPLAT (bit 1) - zbiór przekazany do otworzenia przez pcb_Splatfile i pcb_ConName zostanie zamknięty tylko w przypadku gdy ten bit jest ustawiony.

PRB_NOCLI (bit 2) - standardowo zadanie otwiera się z oknem CLI jeżeli natomiast ustawimy ten bit nigdy nie otrzymamy okna CLI jako właściwego okna Wejścia/Wyjścia.

PRB_INTERACTIVE (bit 3) - ustawia znacznik Interactive w strukturze zadania.

PRB_CODE (bit 4) - jeżeli ustawimy ten znacznik to procedura ASyncRun nie wczyta nowego zadania, ale wykorzysta adres zawarty w komórce pcb_LoadedCode. Możemy dzięki temu uruchamiać tylko części programu jako oddzielne zadania. Zalecane jest jednocześnie ustawienie znacznika PRB_NOCLI i używanie własnych portów do komunikacji z programem, gdyż procedury AmigaDos'u nie są zbyt wygodne do tego celu. Należy tego używać bardzo ostrożnie!

PRB_STDIO (bit 5) - wykreowanie nowego okna konsoli o parametrach podanych w zmiennej pcb_ConName.

Atol - zamiana ciągu Ascii na liczbę

wartość = Atol („Ciąg")

d0 a0

Procedura zamienia ciąg znaków złożony z cyfr na liczbę zawartą w długim słowie.

Wejście:

Ciąg - wskaźnik ciągu znaków zakończony zerem bądź spacją. Liczba może być poprzedzona spacjami.

Wyjście:

Wartość - wartość ciągu znaków. Jeżeli wystąpił błąd to możemy sprawdzić to na dwa sposoby: w języku maszynowym badając znacznik zero (Z), jeżeli został ustawiony to wystąpił błąd. Natomiast - korzystając z języka C - w przypadku błędu zostanie wpisany kod ERRBANDIT do komórki Errno, w przypadku sukcesu komórka ta zostanie wyczyszczona.

Za miesiąc kolejne procedury z tego cyklu.

Marcin „Duddie" Dudar



Witam i proponuję kolejną porcję gier do „nadgryzienia".

SWIV

W czasie gry wciśnij pauze i wpisz „NCC-1701", a uzyskasz nieśmiertelność. Jeżeli nie poskutkuje za pierwszym razem, powtórz operację jeszcze raz.

SUPERCARS

Jako imię pierwszego gracza wpisz „WONDERLAND", by uzyskać zawsze kwalifikację. Jako imię drugiego gracza wpisz „THE SEER" - uzyskasz wszystkie rodzaje broni.

JAMES POND

W czasie gry wpisz „JUNKYARD" i wciśnij Return. Od tej pory jesteś nieśmiertelny. Jeżeli teraz wciśniesz „D" to otworzysz wszystkie przejścia na danym poziomie. Jeśli natomiast wciśniesz jakikolwiek klawisz z dolnego rzędu to przeniesiesz się do następnego etapu.

PREDATOR II

W czasie gry wciśnij pauzę i wpisz „YOU'RE ONE UGLY MOTHER", by uzyskać nieskończoną ilość energii.

ESWAT

Jeśli w czasie gry stwierdzisz, że ubywa ci życia to wciśnij pauzę i wpisz „JUSTIFIED ANCIENTS OF MU MU". Jeśli ekran błyśnie to znaczy, że ten problem masz już z głowy.

Robert „Mr.Raf" Turliński

telegram

Wśród wszystkich, którzy zamówią komplet dysków **PUBLIC DOMAIN PACK** za rok 1991 na **AMIGĘ**

- gwiazdkowa cena 240.000 zł

ROZLOSUJEMY 10 programów D-Mon professional

Do każdego zamówionego zestawu dołączamy gwiazdkowy prezent /gratis/ DISK BOX na 10 dyskietek 3.5''

Wśród wszystkich, którzy zamówią komplet dysków **PUBLIC DOMAIN PACK** za rok 1991 na **C - 64**

- gwiazdkowa cena 200.000 zł

ROZLOSUJEMY 10 programów VOICETRACKER V4.0

Do każdego zamówionego zestawu dołączamy gwiazdkowy prezent /gratis/ DISK BOX na 10 dyskietek 5.25''

Wśród wszystkich, którzy zamówią komplet taśm **PUBLIC DOMAIN PACK** za rok 1991 na **C - 64**

- gwiazdkowa cena 100.000 zł

ROZLOSUJEMY 10 programów VOICETRACKER V4.0

Wszyscy, którzy zakupią **D-Mon professional** w okresie gwiazdkowym otrzymują dodatkowo /gratis/ **MOUSE PAD!**

Nasza oferta gwiazdkowa ważna jest dla wpłat dokonanych od dnia ukazania się niniejszego numeru do 25 stycznia 1992 roku.

z ostatniej chwili

PUBLIC DOMAIN PACK

PUBLIC DOMAIN PACK C-64

Wrzesień

strona A:

- Mega Demo grupy FLASH

strona B:

- Hot Shot - magazyn dyskowy
- Code Sucker monitor - pr. użytkowy grupy PADUA
- Mountain Ride - gra w BASIC

Październik

strona A i B:

- MEGA DEMO „AIRDANCE 4" grupy T.A.T.

PUBLIC DOMAIN PACK TAPE NR 1

- SINUSDATA - EDITOR
- FAST CRUNCHER V3
- ANAL S.C. IBEYOND
- VECTOR - VICTORY
- PUZZLENOID +4
- TUNE OF MONTH #1
- NIM
- STRZAŁKA 64+
- LOGO - WRITER V.2.0
- CAN'T TOUCH IKU!
- INTRO PRV
- BONZIEED !!
- ZAX PACKIS
- READ THIS FIRST
- COMMERCIAL BREAK
- 290 SPRITES!
- NOTE - ABOUT
- BAD NEWS NR2
- TO BAD NEWS...
- CONTACT CORNER!
- PROJEKT DUSZKÓW
- SYMPHONY NR 14
- SYMPHONY NR 15
- SYMPHONY NR 16
- SYMPHONY NR 17
- SYMPHONY NR 18
- SYMPHONY NR 19
- CRUISER/GIANTS
- NOTE > ANO < PADUA
- LET'S DYSP!
- FINALTAPE
- MUSIC - SEARCHER

PUBLIC DOMAIN PACK TAPE NR 2

- TURBO
- PUBL. DOMAIN. INFO
- FONTGRUB 1.0
- DREPTACZ BASIC
- LOAD DIS FIRSY
- MACROASSEMBLER
- TURBOASSEMBLER
- RELOCATOR
- LOGOPAINTER 3!
- REASSEBLER
- SPRITE - EDITOR
- FAST - CRUEL U.2.5
- HIGHLIFE NR5
- AXEL NEWS NR1
- GWIAZDY
- FLIGRAPH 2.2/BML
- NOTE TO FLI V.2.2
- DISKNOTKA/PADUA
- MEGA PACKER/T
- MIST II/ VISION
- TTECHSCR &DYSP
- PLASMA - WORLD
- VECTORBOBS...
- VECTOR - PLOTS
- FLI - UPSCROLL
- BORDER - HIRES
- ROCK AROUND
- FACEWRITER
- CHAR EDIT 2+2
- DISKNOTER
- DESTINATION'91
- CONTACTDEMO/ORE
- FONTEDITOR
- THE END

PUBLIC DOMAIN PACK AMIGA

Lipiec

- Sanity - demo
- Amiga - Tanx (1Mb) - gra
- Little Beau (1MB) - gra
- There is A Light/Tonid - modules

Sierpień

- Real 3D - demo nowego programu do raytraceing'u
- Moduł Muzyczny XTC STEREO

Wrzesień

- MODUŁY MUZYCZNE dla programu TFMX:
  > R - TYPE
  > THE HOUSE OF TECHNO
- VIRUS EXPERT v181 + 143
- BOOT BLOCK'I
  > BOOTX v 3.80
  > IMPLODER v 4.0

Październik

- ANARCHY - „THE INSPIRATION IS NONE"
- DUAL CREW - „NEW DIMENSION"
- SANITY - „ELYSIUM"

Listopad

- COMPUTER HEAD - animacja
- CONFUSED - MODUŁY POD MEDPLAYER I WIELE SAMPLI
- ROCKED -
- SAVE GAME „MONKEY ISLAND"

Spis treści zestawów z poprzednich miesięcy - patrz wcześniejsze numery naszego pisma. Zestawy „64 plus 4 PUBLIC DOMAIN PACK" można zamawiać wpłacając na konto: Bank PKO SA Oddział w Bydgoszczy konto nr:5.09011-400522.7-136-11-111.0 następujących kwot: 20.000zł za pojedyńczy zestaw dla C-64, 25.000zł za zestaw dla AMIGI. Kwoty te obejmują koszt dyskietki, koszty kopiowania, opakowania i przesyłki pocztowej. Blankiety wpłat powinny być **CZYTELNIE** wypełnione i zawierać: **imię i nazwisko, dokładny adres zamawiającego, skrót „PDP-64"**(jeśli zamawiamy zestaw dla C-64) lub **„PDP-A"**(zestaw dla Amigi) - dane te prosimy umieszczać na **wszystkich** odcinkach dowodu wpłaty. W prenumeracie zestawy kosztują: PDP-64 - 18.000zł (12 numerów 216tys zł), PDP-A - 22.000zł (12 numerów 264tys zł). Prenumeratę można zawrzeć w dowolnym terminie na okres od 3 do 12 miesięcy (do końca roku kalendarzowego). Prenumerata może obejmować miesiące od początku roku - tzn. zamawiając całoroczną prenumeratę np. w październiku, w pierwszej przesyłce otrzymacie wszystkie poprzednie zestawy.

ZAMÓW NIE ZWLEKAJ!

GWARANCJĘ SYSTEMATYCZNEGO OTRZYMYWANIA NASZEGO PISMA ZAPEWNIA TYLKO

PRENUMERATA

Wśród wszystkich, którzy do 25 stycznia 1992 roku /decyduje data stempla pocztowego/ zamówią całoroczną prenumeratę

ROZLOSUJEMY

AMIGĘ CDTV!
20 szt. joystick'ów
oraz
100 upominków
/o wartości ok. 20.000 zł każdy/



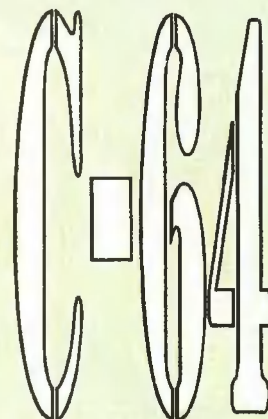
Od stycznia 1992 r. nasze pismo kosztować będzie 10.000 zł. Wszyscy którzy do 25 stycznia wykupią roczną prenumeratę mogą skorzystać ze zniżki wpłacając nie 120.000 zł, a tylko 110.000 zł.

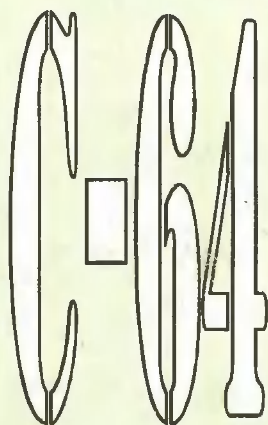
Wpłaty należy przesyłać na konto: BANK PKO S.A. Bydgoszcz, konto nr 5.09011-400522.7-136-11-111.0.

Blankiety wpłat powinny być czytelnie wypełnione i zawierać następujące informacje: imię i nazwisko lub nazwę instytucji, dokładny adres zamawiającego, liczbę zamawianych egzemplarzy oraz okres prenumeraty.

Gwiazdkowe upominki
- szczegóły wewnątrz numeru

VOICETRACKER V4.0





Rewelacyjny program muzyczny!

Tylko **50.000 zł** kosztuje fantastyczny edytor muzyczny wykorzystujący ogromne możliwości dźwiękowe komputera Commodore - 64. Oferowany zestaw zawiera dyskietkę lub taśmę magnetofonową z programem VOICETRACKER V4.0, trzydzieści demonstracji muzycznych, oraz dokładną instrukcję. **UWAGA! Wersja magnetofonowa tylko 40.000zł.!**

Przedsiębiorstwo ABUK posiada wyłączność na dystrybucję tego programu. Wszelkie kopiowanie programu i powielanie instrukcji jest zabronione. Nabywcy otrzymują rejestrowane kopie programu wraz z prawem nabywania nowych wersji po znacznie obniżonych cenach oraz wymiany dyskietki w razie uszkodzenia. Studiom komputerowym proponujemy zakup hurtowy (przy zakupie powyżej 10 kompletów udzielamy 20% rabatu).

Chcąc stać się posiadaczem programu VOICETRACKER V4.0 wystarczy dokonać wpłaty 50.000zł (wersja dyskowa) lub 40.000zł (taśma) na konto: Bank PKO SA Bydgoszcz, konto nr: 5.09011-400522.7-136-11-111.0.

Na blankiecie prosimy czytelnie podać swoje imię, nazwisko i adres wraz z dopiskiem „VV4.0" uzupełnionym literką „T" - taśma lub „D" - dyskietka.

